gost705 и gost705s — стили оформления библиографии в соответствии с ГОСТ Р 7.0.5-2008

Станислав Кручинин

E-mail: stanislav.kruchinin@gmail.com

Версия 1.3.0

2013/12/25

Содержание

1 Исходный текст

```
1
2 ⟨∗gost705 | gost705s⟩
3
```

1.1 Поля, используемые в bib-файлах

```
4 ENTRY
5  { address
6    annote
7    author
8    booktitle
9    bookauthor
10   chapter
11   edition
12   editor
13   compiler
14   howpublished
15   institution
```

```
16    journal
17    key
18    month
19    note
20    number
21    organization
22    pages
23    numpages
24    publisher
25    school
26    series
27    title
28    type
29    volume
30    year
31    language
32    booklanguage
33    url
34    urldate
35    isbn
36    doi
37    archive
38    archivePrefix
39    primaryClass
40    eprint
41    authortype
42    media
43    location
44    ipc
45    holder
46    credits
47    reqnumber
48    reqdate
49    pubdate
50    publication
51    prdate
52    prnumber
53    prcountry
54    pagetotal
55  }
56  {}
57  { label }
58
```

1.2 Объявление и инициализация переменных и констант

```
59 INTEGERS { output.state before.all mid.sentence after.sentence after.block
60 after.dblslash after.slash after.colon after.semicolon after.dash }
61
```

init.state.consts

```
62 FUNCTION {init.state.consts}
63 {
64   #0 'before.all :=
65   #1 'mid.sentence :=
66   #2 'after.sentence :=
67   #3 'after.block :=
68   #4 'after.dblslash :=
69   #5 'after.slash :=
70   #6 'after.colon :=
71   #7 'after.semicolon :=
72   #8 'after.dash :=
73 }
74
75 STRINGS { s t }
76
77 STRINGS { curlanguage }
```

2

1.3 Переключение языка

change.language

```
79 FUNCTION {change.language}
80 {
81   booklanguage empty$
82     { "" }
83     { booklanguage  'curlanguage :=
84       "\selectlanguageifdefined{"
85       curlanguage *
86       "}" *
87     }
88   if$
89 }
90
```

1.4 Вывод библиографической записи

output.nonnull

```
91  FUNCTION {output.nonnull}
92  {
93    swap$
94    output.state mid.sentence =
95      { ", " * write$ }
96      { output.state after.block =
97          { add.period$ " " * write$
98            newline$
99            "\newblock " write$
100         }
101         { output.state before.all =
102             'write$
103             { output.state after.dblslash =
104                 { "~//" * change.language * " " * write$ }
105                 { output.state after.slash =
106                     { "~/ " * write$ }
107                     { output.state after.colon =
108                         { ": " * write$ }
109                         { output.state after.semicolon =
110                             { "; " * write$ }
111                             { output.state after.dash =
112                                 { "~--- " * write$ }
113                                 { add.period$ " " * write$ }
114                               if$
115                             }
116                           if$
117                         }
118                       if$
119                     }
120                   if$
121                 }
122               if$
123             }
124           if$
125         }
126       if$
127       mid.sentence 'output.state :=
128     }
129   if$
130 }
131
```

output

```
132 FUNCTION {output}
133 {
```

```
134   duplicate$ empty$
135     'pop$
136     'output.nonnull
137   if$
138 }
139
```

`output.check`

```
140 FUNCTION {output.check}
141 {
142   't :=
143   duplicate$ empty$
144     { pop$ "empty " t * " in " * cite$ * warning$ }
145     'output.nonnull
146   if$
147 }
148
```

1.5   Вывод блоков и ограничителей

`fin.entry`

```
149 FUNCTION {fin.entry}
150 {
151   add.period$
152   write$
153   newline$
154 }
155
```

`new.block`

```
156 FUNCTION {new.block}
157 {
158   output.state before.all =
159     'skip$
160     { after.block 'output.state := }
161   if$
162 }
163
```

`new.dblslash`

```
164 FUNCTION {new.dblslash}
165 {
166   output.state before.all =
167     'skip$
168     { after.dblslash 'output.state := }
169   if$
170 }
171
```

`new.slash`

```
172 FUNCTION {new.slash}
173 {
174   output.state before.all =
175     'skip$
176     { after.slash 'output.state := }
177   if$
178 }
179
```

`new.colon`

```
180 FUNCTION {new.colon}
181 {
182   output.state before.all =
183     'skip$
184     { after.colon 'output.state := }
185   if$
186 }
187
```

new.semicolon
```
188 FUNCTION {new.semicolon}
189 {
190   output.state before.all =
191     'skip$
192     { after.semicolon 'output.state := }
193   if$
194 }
195
```

new.dash
```
196 FUNCTION {new.dash}
197 {
198   output.state before.all =
199     'skip$
200     { after.dash 'output.state := }
201   if$
202 }
203
```

new.sentence
```
204 FUNCTION {new.sentence}
205 {
206   output.state after.block =
207     'skip$
208     { output.state before.all =
209        'skip$
210        { after.sentence 'output.state := }
211      if$
212    }
213   if$
214 }
215
```

add.blank
```
216 FUNCTION {add.blank} { " " * before.all 'output.state := }
217
```

not
```
218 FUNCTION {not}
219 {
220    { #0 }
221    { #1 }
222   if$
223 }
224
```

and
```
225 FUNCTION {and}
226 {
227    'skip$
228    { pop$ #0 }
229   if$
230 }
231
```

or
```
232 FUNCTION {or}
233 {
234    { pop$ #1 }
235    'skip$
236   if$
237 }
238
```

non.stop
```
239 FUNCTION {non.stop}
240 {
241   duplicate$
```

```
242   "}" * add.period$
243   #-1 #1 substring$ "." =
244 }
245
```

new.block.checka

```
246 FUNCTION {new.block.checka}
247 {
248   empty$
249     'skip$
250     'new.block
251   if$
252 }
253
```

new.block.checkb

```
254 FUNCTION {new.block.checkb}
255 {
256   empty$
257   swap$ empty$
258   and
259     'skip$
260     'new.block
261   if$
262 }
263
```

new.sentence.checka

```
264 FUNCTION {new.sentence.checka}
265 {
266   empty$
267     'skip$
268     'new.sentence
269   if$
270 }
271
```

new.sentence.checkb

```
272 FUNCTION {new.sentence.checkb}
273 {
274   empty$
275   swap$ empty$
276   and
277     'skip$
278     'new.sentence
279   if$
280 }
281
```

field.or.null

```
282 FUNCTION {field.or.null}
283 {
284   duplicate$ empty$
285     { pop$ "" }
286     'skip$
287   if$
288 }
289
```

emphasize

```
290 FUNCTION {emphasize}
291 {
292   duplicate$ empty$
293     { pop$ "" }
294     { "\BibEmph{" swap$ * "}" * }
295   if$
296 }
297
```

```
298 FUNCTION {space.word} { " " swap$ * " " * }
299
```

1.6 Общепринятые сокращения

```
300 FUNCTION {bbl.edby} % { "\bbledby{}" }
301 {
302   curlanguage "english" =
303   { "Ed. by " }
304     { curlanguage "german" =
305       { "ed. " }
306         { curlanguage "russian" =
307           { "{\cyr\CYRP\cyro\cyrd\ \cyrr\cyre\cyrd.\ }" }
308           { "language is not defined: " language  "edited by" * *
309            warning$ "edited by" }
310         if$
311         }
312       if$
313       }
314   if$
315 }
316
```

```
317 FUNCTION {bbl.cmplr}
318 {
319   curlanguage "english" =
320   { "Compiler " }
321     { curlanguage "german" =
322       { "Hrsg. " }
323         { curlanguage "russian" =
324         { "{\cyr\CYRS\cyro\cyrs\cyrt.\ }" }
325         { "language is not defined: " language  "compiler" * * warning$ "compiler" }
326       if$
327       }
328     if$
329     }
330   if$
331 }
332
```

```
333 FUNCTION {bbl.edition} % { "\bbledition{}" }
334 {
335   curlanguage "english" =
336    { "edition" }
337    { curlanguage "german" =
338      { "Aufl." }
339        { curlanguage "russian" =
340          { "{\cyr\cyri\cyrz\cyrd.}" }
341          { "language is not defined: " language  "edition" * * warning$ "edition" }
342        if$
343      }
344     if$
345     }
346   if$
347 }
348
```

```
349 FUNCTION {bbl.vvolume} % { "\bblVolume{}" }
350 {
351   curlanguage "english" =
352     { "Volume" }
353     { curlanguage "german" =
354         { "Band" }
```

```
355        { curlanguage "russian" =
356          { "{\cyr\CYRT\cyro\cyrm}" }
357          { "language is not defined: " language  "Volume" * * warning$ "Volume" }
358        if$
359        }
360      if$
361      }
362    if$
363 }
364
```

bbl.vvol

```
365 FUNCTION {bbl.vvol} % { "\bblVol{}" }
366 {
367   curlanguage "english" =
368     { "Vol." }
369     { curlanguage "german" =
370       { "Bd." }
371       { curlanguage "russian" =
372         { "{\cyr\CYRT.}" }
373         { "language is not defined: " language  "Vol." * * warning$ "Vol." }
374       if$
375       }
376     if$
377     }
378   if$
379 }
380
```

bbl.iissue

```
381 FUNCTION {bbl.iissue} % { "\bblIssue{}" }
382 {
383   curlanguage "english" =
384     { "Issue" }
385     { curlanguage "german" =
386       { "Heft" }
387       { curlanguage "russian" =
388         { "{\cyr\CYRV\cyrery\cyrp\cyru\cyrs\cyrk}" }
389         { "language is not defined: " language * warning$ "Issue" }
390       if$
391       }
392     if$
393     }
394   if$
395 }
396
```

bbl.iiss

```
397 FUNCTION {bbl.iiss} % { "\bblIss{}" }
398 {
399   curlanguage "english" =
400     { "Iss." }
401     { curlanguage "german" =
402       { "H." }
403       { curlanguage "russian" =
404         { "{\cyr\CYRV\cyrery\cyrp.}" }
405         { "language is not defined: " language * warning$ "Iss." }
406       if$
407       }
408     if$
409     }
410   if$
411 }
412
```

bbl.of

```
413 FUNCTION {bbl.of} % { "\bblof{}" }
414 {
415   curlanguage "english" =
```

```
416        { "of" }
417        { curlanguage "german" =
418          { "von" }
419          { curlanguage "russian" =
420            { "{\cyr\cyri\cyrz}" }
421            { "language is not defined: " language * warning$ "of" }
422          if$
423          }
424        if$
425        }
426    if$
427 }
428
```

```
429 FUNCTION {bbl.etal} % { "\bblof{}" }
430 {
431    curlanguage "english" =
432      { "et~al." }
433      { curlanguage "german" =
434        { "u.~a." }
435        { curlanguage "russian" =
436          { "{\cyr\cyri~\cyrd\cyrr.}" }
437          { "language is not defined: " language * warning$ "etal" }
438        if$
439        }
440      if$
441      }
442    if$
443 }
444
```

```
445 FUNCTION {bbl.nnumber} % { "\bblNumber{}" }
446 {
447    curlanguage "english" =
448      { "Number" }
449      { curlanguage "german" =
450        { "Heft" }
451        { curlanguage "russian" =
452          { "{\cyr\CYRN\cyro\cyrm\cyre\cyrr}" }
453          { "language is not defined: " language * warning$ "Number" }
454        if$
455        }
456      if$
457      }
458    if$
459 }
460
```

```
461 FUNCTION {bbl.number} % { "\bblnumber{}" }
462 {
463    curlanguage "english" =
464      { "number" }
465      { curlanguage "german" =
466        { "Heft" }
467        { curlanguage "russian" =
468          { "{\cyr\cyrn\cyro\cyrm\cyre\cyrr}" }
469          { "language is not defined: " language * warning$ "number" }
470        if$
471        }
472      if$
473      }
474    if$
475 }
476
```

```
477 FUNCTION {bbl.nr} % { "\bblno{}" }
478 {
479   curlanguage "english" =
480     { "no." }
481     { curlanguage "german" =
482       { "H." }
483       { curlanguage "russian" =
484         { "{\cyr\textnumero}" }
485         { "language is not defined: " language * warning$ "no" }
486       if$
487       }
488     if$
489     }
490   if$
491 }
492
```

bbl.nnr

```
493 FUNCTION {bbl.nnr} % { "\bblno{}" }
494 {
495   curlanguage "english" =
496     { "No." }
497     { curlanguage "german" =
498       { "H." }
499       { curlanguage "russian" =
500         { "{\cyr\textnumero}" }
501         { "language is not defined: " language * warning$ "No" }
502       if$
503       }
504     if$
505     }
506   if$
507 }
508
```

bbl.in

```
509 FUNCTION {bbl.in} % { "\bblin{}" }
510 {
511   curlanguage "english" =
512     { "in" }
513     { curlanguage "german" =
514       { "in" }
515       { curlanguage "russian" =
516         { "{\cyr\cyrv}" }
517         { "language is not defined: " language * warning$ "in" }
518       if$
519       }
520     if$
521     }
522   if$
523 }
524
```

bbl.iin

```
525 FUNCTION {bbl.iin} % { "\bblIn{}" }
526 {
527   curlanguage "english" =
528     { "In" }
529     { curlanguage "german" =
530       { "In" }
531       { curlanguage "russian" =
532         { "{\cyr\CYRV}" }
533         { "language is not defined: " language * warning$ "In" }
534       if$
535       }
536     if$
537     }
538   if$
```

```
539 }
540
```

bbl.pages
```
541 FUNCTION {bbl.pages} % { "\bblpp." }
542 {
543   curlanguage "english" =
544     { "p." }
545     { curlanguage "german" =
546       { "S." }
547       { curlanguage "russian" =
548         { "{\cyr\cyrs.}" }
549         { "language is not defined: " language * warning$ "p." }
550       if$
551       }
552     if$
553     }
554   if$
555 }
556
```

bbl.page
```
557 FUNCTION {bbl.page} % { "\bblp." }
558 {
559   curlanguage "english" =
560     { "p." }
561     { curlanguage "german" =
562       { "S." }
563       { curlanguage "russian" =
564         { "{\cyr\cyrs.}"}
565         { "language is not defined: " language * warning$ "p." }
566       if$
567       }
568     if$
569     }
570   if$
571 }
572
```

bbl.ppages
```
573 FUNCTION {bbl.ppages} % { "\bblPp." }
574 {
575   curlanguage "english" =
576     { "P." }
577     { curlanguage "german" =
578       { "S." }
579       { curlanguage "russian" =
580         { "{\cyr\CYRS.}" }
581         { "language is not defined: " language * warning$ "P." }
582       if$
583       }
584     if$
585     }
586   if$
587 }
588
```

bbl.ppage
```
589 FUNCTION {bbl.ppage} % { "\bblP." }
590 {
591   curlanguage "english" =
592     { "P." }
593     { curlanguage "german" =
594       { "S." }
595       { curlanguage "russian" =
596         { "{\cyr\CYRS.}" }
597         { "language is not defined: " language * warning$ "P." }
598       if$
599       }
```

```
                600       if$
                601     }
                602   if$
                603 }
                604

bbl.techrep
                605 FUNCTION {bbl.techrep}
                606 {
                607   curlanguage "english" =
                608     { "Tech.~Rep." }
                609     { curlanguage "german" =
                610       { "Tech.~Bericht" }
                611       { curlanguage "russian" =
                612         { "{\cyr\CYRT\cyre\cyrh.\ \cyrd\cyro\cyrk\cyrl\cyra\cyrd.}" }
                613         { "language is not defined: " language * warning$ "Tech Rep" }
                614       if$
                615       }
                616     if$
                617     }
                618   if$
                619 }
                620

bbl.mthesis
                621 FUNCTION {bbl.mthesis}
                622 {
                623   curlanguage "english" =
                624     { "Master's thesis" }
                625     { curlanguage "german" =
                626       { "Diss.~Mag." }
                627       { curlanguage "russian" =
                628         { "{\cyr\CYRM\cyra\cyrg\cyri\cyrs\cyrt\cyre\cyrr\cyrs\cyrk\cyra\cyrya\ "
                629           "\cyrd\cyri\cyrs\cyrs\cyre\cyrr\cyrt\cyra\cyrc\cyri\cyrya}" * }
                630         { "language is not defined: " language * warning$ "Master thesis" }
                631       if$
                632       }
                633     if$
                634     }
                635   if$
                636 }
                637

bbl.phdthesis
                638 FUNCTION {bbl.phdthesis}
                639 {
                640   curlanguage "english" =
                641     { "Ph.\,D. thesis" }
                642     { curlanguage "german" =
                643       { "Diss.~Ph.\,D." }
                644       { curlanguage "russian" =
                645         { "{\cyr\CYRK\cyra\cyrn\cyrd\cyri\cyrd\cyra\cyrt\cyrs\cyrk\cyra\cyrya\ "
                646           "\cyrd\cyri\cyrs\cyrs\cyre\cyrr\cyrt\cyra\cyrc\cyri\cyrya}" * }
                647         { "language is not defined: " language * warning$ "PhD thesis" }
                648       if$
                649       }
                650     if$
                651     }
                652   if$
                653 }
                654

bbl.docdisser
                655 FUNCTION {bbl.docdisser}
                656 {
                657   curlanguage "english" =
                658     { "Dr.\,Sci. dissertation" }
                659     { curlanguage "german" =
                660       { "Diss.~Dr." }
```

12

```
661       { curlanguage "russian" =
662         { "{\cyr\CYRD\cyro\cyrk\cyrt\cyro\cyrr\cyrs\cyrk\cyra\cyrya\ "
663           "\cyrd\cyri\cyrs\cyrs\cyre\cyrr\cyrt\cyra\cyrc\cyri\cyrya}" * }
664         { "language is not defined: " language * warning$ "Dr. Sci. dissertation" }
665       if$
666       }
667     if$
668     }
669   if$
670 }
671
```

bbl.req

```
672 FUNCTION {bbl.req}
673 {
674   curlanguage "english" =
675     { "req." }
676     { curlanguage "german" =
677       { "ang." }
678       { curlanguage "russian" =
679         { "{\cyr\cyrz\cyra\cyrya\cyrv\cyrl.}" }
680         { "language is not defined: " language * warning$ "req" }
681       if$
682       }
683     if$
684     }
685   if$
686 }
687
```

bbl.publ

```
688 FUNCTION {bbl.publ}
689 {
690   curlanguage "english" =
691     { "publ." }
692     { curlanguage "german" =
693       { "ausg." }
694       { curlanguage "russian" =
695         { "{\cyr\cyro\cyrp\cyru\cyrb\cyrl.}" }
696         { "language is not defined: " language * warning$ "In" }
697       if$
698       }
699     if$
700     }
701   if$
702 }
703
```

bbl.prio

```
704 FUNCTION {bbl.prio}
705 {
706   curlanguage "english" =
707     { "priority" }
708     { curlanguage "german" =
709       { "Prioritat" }
710       { curlanguage "russian" =
711         { "{\cyr\cyrp\cyrr\cyri\cyro\cyrr\cyri\cyrt\cyre\cyrt}" }
712         { "language is not defined: " language * warning$ "In" }
713       if$
714       }
715     if$
716     }
717   if$
718 }
719
```

1.7 Формат элементов библиографических ссылок

```
720 INTEGERS { nameptr namesleft numnames }
```

13

```
721
722 FUNCTION {format.names}
723 {
724   's :=
725   #1 'nameptr :=
726   s num.names$ 'numnames :=
727   numnames 'namesleft :=
728     { namesleft #0 > }
729     { s nameptr
730       "{vv~}{ll}{~f.}{, jj}" format.name$ 't :=
731       nameptr #1 >
732         { nameptr #4 =
733           numnames #4 > and
734             { "others" 't :=
735               #1 'namesleft := }
736             'skip$
737           if$
738           namesleft #1 >
739             { ", " * t * }
740             { t "others" =
741               t "~others" =
742               or
743                 { " " * bbl.etal * }
744                 { ", " * t * }
745               if$
746             }
747           if$
748         }
749         't
750       if$
751       nameptr #1 + 'nameptr :=
752       namesleft #1 - 'namesleft :=
753     }
754   while$
755 }
756
```

```
757 FUNCTION {format.names.rev}
758 {
759   's :=
760   #1 'nameptr :=
761   s num.names$ 'numnames :=
762   numnames 'namesleft :=
763     { namesleft #0 > }
764     { s nameptr
765       "{f.}{~vv}{~ll}{, jj}" format.name$ 't :=
766       nameptr #1 >
767         { nameptr #4 =
768           numnames #4 > and
769             { "others" 't :=
770               #1 'namesleft := }
771             'skip$
772           if$
773           namesleft #1 >
774             { ", " * t * }
775             { t "others" =
776               t "~others" =
777               or
778                 { " " * bbl.etal * }
779                 { ", " * t * }
780               if$
781             }
782           if$
783         }
```

14

```
784          't
785        if$
786        nameptr #1 + 'nameptr :=
787        namesleft #1 - 'namesleft :=
788      }
789    while$
790 }
791
```

format.authors

```
792 FUNCTION {format.authors}
793 {
794    author empty$
795      { "" }
796      { author format.names emphasize }
797    if$
798 }
799
```

format.bookauthors

```
800 FUNCTION {format.bookauthors}
801 {
802    bookauthor empty$
803      { "" }
804      { bookauthor format.names }
805    if$
806 }
807
```

format.authors.after

```
808 FUNCTION {format.authors.after}
809 {
810    author empty$
811      { "" }
812      { author format.names.rev }
813    if$
814 }
815
```

format.bookauthors.after

```
816 FUNCTION {format.bookauthors.after}
817 {
818    bookauthor empty$
819      { "" }
820      { bookauthor format.names.rev }
821    if$
822 }
823
```

format.editors.after

```
824 FUNCTION {format.editors.after}
825 {
826    editor empty$
827      { "" }
828      { bbl.edby editor format.names.rev * }
829    if$
830 }
831
```

format.compiler.after

```
832 FUNCTION {format.compiler.after}
833 {
834    compiler empty$
835      { "" }
836      { bbl.cmplr compiler format.names.rev * }
837    if$
838 }
839
```

format.title

```
840 FUNCTION {format.title}
```

```
841 {
842   title empty$
843     { "" }
844     { title }
845   if$
846 }
847
```

format.date

```
848 FUNCTION {format.date}
849 {
850   year empty$
851     { month empty$
852         { "" }
853         { "there's a month but no year in " cite$ * warning$ month }
854       if$
855     }
856     { month empty$
857         'year
858         { year ". " quote$ "--- " month * * * * }
859       if$
860     }
861   if$
862 }
863
```

output.address.publisher.date

```
864 FUNCTION {output.address.publisher.date}
865 {
866   address empty$
867     'skip$
868     { address output
869       publisher empty$
870         'skip$
871         { new.colon }
872       if$
873     }
874   if$
875   publisher output
876   format.date output
877 }
878
```

output.bibitem

```
879 FUNCTION {output.bibitem}
880 {
881   newline$
882   "\bibitem" "{" * write$
883   cite$ write$
884   "}" write$
885   newline$
886   language empty$
887     { "english" 'curlanguage := }
888     { language  'curlanguage := }
889   if$
890   "\selectlanguageifdefined{" write$
891   curlanguage write$
892   "}" write$ newline$
893   ""
894   before.all 'output.state :=
895 }
896
```

n.dashify

```
897 FUNCTION {n.dashify}
898 {
899   't :=
900   ""
901     { t empty$ not }
```

16

```
902    { t #1 #1 substring$ "-" =
903        { t #1 #2 substring$ "--" = not
904            { "--" *
905                t #2 global.max$ substring$ 't :=
906            }
907            {    { t #1 #1 substring$ "-" = }
908                { "-" *
909                    t #2 global.max$ substring$ 't :=
910                }
911                while$
912            }
913          if$
914        }
915        { t #1 #1 substring$ *
916            t #2 global.max$ substring$ 't :=
917        }
918      if$
919    }
920  while$
921 }
922
```

```
923 FUNCTION {word.in} { bbl.iin " " * }
924
```

```
925 FUNCTION {format.btitle} { title }
926
```

```
927 FUNCTION {tie.or.space.connect}
928 {
929   duplicate$ text.length$ #3 <
930     { "~" }
931     { " " }
932   if$
933   swap$ * *
934 }
935
```

```
936 FUNCTION {tie.connect} { "~" swap$ * * }
937
```

```
938 FUNCTION {either.or.check}
939 {
940   empty$
941     'pop$
942     { "can't use both " swap$ * " fields in " * cite$ * warning$ }
943   if$
944 }
945
```

```
946 FUNCTION {format.bvolume}
947 {
948   volume empty$
949     { "" }
950     { bbl.vvol volume tie.connect
951       series empty$
952         'skip$
953         { bbl.of space.word * series emphasize * }
954       if$
955       "volume and number" number either.or.check
956     }
957   if$
958 }
959
```

```
960 FUNCTION {format.number.series}
961 {
962   volume empty$
963     { number empty$
964         { series field.or.null }
965         { series empty$
966             { "there's a number but no series in " cite$ * warning$
967               bbl.nnr
968             }
969             { new.sentence
970               bbl.iiss series tie.or.space.connect
971               ", " * bbl.nr
972               tie.or.space.connect
973             }
974           if$
975           number tie.or.space.connect
976         }
977       if$
978     }
979     { "" }
980   if$
981 }
982
```

```
983 FUNCTION {is.num}
984 {
985   chr.to.int$
986   duplicate$ "0" chr.to.int$ < not
987   swap$ "9" chr.to.int$ > not and
988 }
989
```

```
990 FUNCTION {extract.num}
991 {
992   duplicate$ 't :=
993   "" 's :=
994     { t empty$ not }
995     { t #1 #1 substring$
996       t #2 global.max$ substring$ 't :=
997       duplicate$ is.num
998         { s swap$ * 's := }
999         { pop$ "" 't := }
1000      if$
1001    }
1002  while$
1003  s empty$
1004    'skip$
1005    { pop$ s }
1006  if$
1007 }
1008
```

```
1009 FUNCTION {convert.edition} { edition }
1010
```

```
1011 FUNCTION {format.edition}
1012 {
1013   edition empty$
1014     { "" }
1015     { output.state mid.sentence =
1016         { convert.edition "l" change.case$ " " * bbl.edition * }
1017         { convert.edition "t" change.case$ " " * bbl.edition * }
1018       if$
1019     }
```

18

```
                        1020   if$
                        1021 }
                        1022
                        1023 INTEGERS { multiresult }
                        1024

   multi.page.check
                        1025 FUNCTION {multi.page.check}
                        1026 {
                        1027   't :=
                        1028   #0 'multiresult :=
                        1029     { multiresult not
                        1030       t empty$ not
                        1031       and
                        1032     }
                        1033     { t #1 #1 substring$
                        1034       duplicate$ "-" =
                        1035       swap$ duplicate$ "," =
                        1036       swap$ "+" =
                        1037       or or
                        1038         { #1 'multiresult := }
                        1039         { t #2 global.max$ substring$ 't := }
                        1040       if$
                        1041     }
                        1042   while$
                        1043   multiresult
                        1044 }
                        1045

      format.pages
                        1046 FUNCTION {format.pages}
                        1047 {
                        1048   pages empty$
                        1049     { "" }
                        1050     { pages multi.page.check
                        1051         { bbl.ppages pages n.dashify tie.connect }
                        1052         { bbl.ppage pages tie.connect }
                        1053       if$
                        1054     }
                        1055   if$
                        1056 }
                        1057

 format.pages.page
                        1058 FUNCTION {format.pages.page}
                        1059 {
                        1060   pages empty$
                        1061     { numpages empty$
                        1062         { "" }
                        1063         { numpages bbl.pages tie.connect }
                        1064       if$
                        1065     }
                        1066     { format.pages }
                        1067   if$
                        1068 }
                        1069

format.vol.num.pages
                        1070 FUNCTION {format.vol.num.pages}
                        1071 {
                        1072   volume field.or.null
                        1073   number empty$
                        1074     'skip$
                        1075     { ", no." number tie.or.space.connect *
                        1076       volume empty$
                        1077         { "there's a number but no volume in " cite$ * warning$ }
                        1078         'skip$
                        1079       if$
                        1080     }
```

```
                     1081  if$
                     1082  pages empty$
                     1083    'skip$
                     1084    { duplicate$ empty$
                     1085       { pop$ format.pages }
                     1086       { ": " * pages n.dashify * }
                     1087      if$
                     1088    }
                     1089  if$
                     1090 }
                     1091
```

format.volume
```
                     1092 FUNCTION {format.volume}
                     1093 {
                     1094   volume empty$
                     1095     { "" }
                     1096     { bbl.vvol volume tie.or.space.connect }
                     1097   if$
                     1098 }
                     1099
```

format.number
```
                     1100 FUNCTION {format.number}
                     1101 { number empty$
                     1102     { "" }
                     1103     { bbl.nr number tie.or.space.connect }
                     1104   if$
                     1105 }
                     1106
```

empty.misc.check
```
                     1107 FUNCTION {empty.misc.check}
                     1108 {
                     1109   author empty$ title empty$ howpublished empty$
                     1110   month empty$ year empty$ note empty$
                     1111   and and and and and
                     1112   key empty$ not and
                     1113     { "all relevant fields are empty in " cite$ * warning$ }
                     1114     'skip$
                     1115   if$
                     1116 }
                     1117
```

format.thesis.type
```
                     1118 FUNCTION {format.thesis.type}
                     1119 {
                     1120   type empty$
                     1121     'skip$
                     1122     { pop$
                     1123       type "t" change.case$
                     1124     }
                     1125   if$
                     1126 }
                     1127
```

format.tr.number
```
                     1128 FUNCTION {format.tr.number}
                     1129 {
                     1130   type empty$
                     1131     { bbl.techrep }
                     1132     'type
                     1133   if$
                     1134   number empty$
                     1135     { "t" change.case$ }
                     1136     { number tie.or.space.connect }
                     1137   if$
                     1138 }
                     1139
```

```
         author.before
                     1140 FUNCTION {author.before}
                     1141 {
                     1142   author empty$
                     1143     'skip$
                     1144     { format.authors output
                     1145       new.sentence
                     1146     }
                     1147   if$
                     1148 }
                     1149

      bookauthor.before
                     1150 FUNCTION {bookauthor.before}
                     1151 {
                     1152   bookauthor empty$
                     1153     'skip$
                     1154     { format.bookauthors output
                     1155       new.sentence
                     1156     }
                     1157   if$
                     1158 }
                     1159

          author.after
                     1160 FUNCTION {author.after}
                     1161 {
                     1162   author empty$
                     1163     'skip$
                     1164     { author num.names$ #3 >
                     1165       { format.authors.after output
                     1166         new.semicolon
                     1167       }
                     1168       'skip$
                     1169     if$
                     1170     }
                     1171   if$
                     1172 }
                     1173

      bookauthor.after
                     1174 FUNCTION {bookauthor.after}
                     1175 {
                     1176   bookauthor empty$
                     1177     'skip$
                     1178     { bookauthor num.names$ #3 >
                     1179       { format.bookauthors.after output
                     1180         new.semicolon
                     1181       }
                     1182       'skip$
                     1183     if$
                     1184     }
                     1185   if$
                     1186 }
                     1187

editor.organization.after
                     1188 FUNCTION {editor.organization.after}
                     1189 {
                     1190   compiler empty$
                     1191     'skip$
                     1192     { format.compiler.after output
                     1193       new.semicolon
                     1194     }
                     1195   if$
                     1196   editor empty$
                     1197     'skip$
                     1198     { format.editors.after output
                     1199       new.semicolon
```

21

```
1200        }
1201    if$
1202    organization empty$
1203      'skip$
1204      { organization output
1205        new.semicolon
1206      }
1207    if$
1208 }
1209
```

format.annote

```
1210 FUNCTION {format.annote}
1211 {
1212   annote empty$
1213     { "" }
1214     { after.sentence 'output.state :=
1215       "\BibAnnote{" annote add.period$ * "}" *
1216     }
1217   if$
1218 }
1219
```

1.8 Создание гиперссылок

format.url

```
1220 FUNCTION {format.url}
1221 {
1222   url empty$
1223     { "" }
1224     { "\BibUrl{" url * "}" * }
1225   if$
1226 }
1227
```

format.urldate

```
1228 FUNCTION {format.urldate}
1229 {
1230   urldate empty$
1231     { "" }
1232     { add.blank
1233       "\BibUrlDate{" urldate * "}" *
1234     }
1235   if$
1236 }
1237
```

output.url

```
1238 FUNCTION {output.url}
1239 {
1240   url empty$
1241     'skip$
1242     { format.url output
1243       format.urldate output
1244     }
1245   if$
1246 }
1247
```

format.isbn

```
1248 FUNCTION {format.isbn}
1249 {
1250   isbn empty$
1251     { "" }
1252     { "ISBN:~\href{http://isbndb.com/search-all.html?kw=" isbn *
1253       "}{" * isbn * "}" *
1254     }
1255   if$
```

```
1256 }
1257
```

```
1258 FUNCTION {add.doi.url}
1259 { duplicate$ empty$
1260    'skip$
1261    { doi empty$
1262      { url empty$
1263         'skip$
1264         { "\href{" url * "}{" * swap$ * "}" * }
1265       if$
1266      }
1267      { "\href{http://dx.doi.org/" doi * "}{" * swap$ * "}" * }
1268      if$
1269    }
1270   if$
1271 }
1272
```

```
1273 FUNCTION {format.eprint}
1274 { eprint empty$
1275    { "" }
1276    { primaryClass empty$
1277      { " \href{http://arxiv.org/abs/" eprint * "}" *
1278        "{" * eprint * "}" *
1279      }
1280      { archivePrefix empty$
1281        { " \href{http://arxiv.org/abs/" eprint * "}" *
1282          "{" * primaryClass * "/" * eprint * "}" *
1283        }
1284        { " \href{http://arxiv.org/abs/" eprint * "}" *
1285        "{" * archivePrefix * ":" * primaryClass * "/" * eprint * "}" *
1286        }
1287       if$
1288      }
1289     if$
1290    }
1291   if$
1292 }
1293
```

```
1294 FUNCTION {add.number}
1295 { duplicate$ empty$
1296    { "" }
1297    { bbl.nr swap$ tie.or.space.connect }
1298   if$
1299 }
1300
```

```
1301 FUNCTION {in.square.brackets}
1302 {
1303   duplicate$ empty$
1304    { pop$ "" }
1305    { "[" swap$ * "]" * }
1306   if$
1307 }
1308
```

```
1309 FUNCTION {in.round.brackets}
1310 {
1311   duplicate$ empty$
1312    { pop$ "" }
1313    { "(" swap$ * ")" * }
```

```
1314    if$
1315 }
1316
```

```
1317 FUNCTION {format.type.number}
1318 {
1319   type empty$
1320     { "" }
1321     {
1322       number empty$
1323       { "" }
1324       { type number tie.or.space.connect }
1325     if$
1326     }
1327   if$
1328 }
1329
```

```
1330 FUNCTION {format.reqdate}
1331 { reqdate empty$
1332     { "" }
1333     { bbl.req reqdate tie.or.space.connect }
1334   if$
1335 }
1336
```

```
1337 FUNCTION {format.pubdate}
1338 { pubdate empty$
1339     { "" }
1340     { bbl.publ pubdate tie.or.space.connect }
1341   if$
1342 }
1343
```

```
1344 FUNCTION {format.prdate}
1345 { prdate empty$
1346     { "" }
1347     { bbl.prio prdate tie.or.space.connect }
1348   if$
1349 }
1350
```

1.9 Описание типов библиографических ссылок

```
1351 FUNCTION {article}
1352 {
1353   output.bibitem
1354   author.before
1355   format.title "title" output.check
1356   new.dblslash
1357   journal emphasize add.doi.url "journal" output.check
1358   new.block
1359   format.date output
1360   new.block
1361   format.volume output
1362   format.number output
1363   new.block
1364   format.pages.page output
1365   new.block
1366   note output
1367   new.sentence
1368   format.eprint output
```

24

```
1369    new.block
1370    output.url
1371    format.annote output
1372    fin.entry
1373 }
1374
```

book

```
1375 FUNCTION {book}
1376 {
1377    output.bibitem
1378    author.before
1379    format.btitle add.doi.url "title" output.check
1380    new.slash
1381    editor.organization.after
1382    new.sentence
1383    format.number.series output
1384    new.block
1385    format.edition output
1386    new.block
1387    output.address.publisher.date
1388    new.block
1389    format.bvolume output
1390    new.block
1391    format.pages.page output
1392    new.block
1393    format.isbn output
1394    new.block
1395    note output
1396    new.sentence
1397    format.eprint output
1398    new.block
1399    output.url
1400    format.annote output
1401    fin.entry
1402 }
1403
```

booklet

```
1404 FUNCTION {booklet}
1405 {
1406    output.bibitem
1407    author.before
1408    format.title "title" output.check
1409    editor.organization.after
1410    new.block
1411    howpublished output
1412    address output
1413    format.date output
1414    new.block
1415    note output
1416    new.sentence
1417    format.eprint output
1418    new.block
1419    output.url
1420    format.annote output
1421    fin.entry
1422 }
1423
```

inbook

```
1424 FUNCTION {inbook}
1425 {
1426    output.bibitem
1427    author.before
1428    format.btitle add.doi.url "title" output.check
1429    new.dblslash
```

```
1430    bookauthor.before
1431    booktitle "booktitle" output.check
1432    editor.organization.after
1433    new.block
1434    format.edition output
1435    new.block
1436    format.number.series output
1437    new.sentence
1438    output.address.publisher.date
1439    new.block
1440    format.bvolume output
1441    new.block
1442    format.pages.page output
1443    new.block
1444    format.isbn output
1445    new.block
1446    note output
1447    new.sentence
1448    format.eprint output
1449    new.block
1450    output.url
1451    format.annote output
1452    fin.entry
1453 }
1454
```

```
1455 FUNCTION {incollection}
1456 {
1457    output.bibitem
1458    author.before
1459    new.sentence
1460    format.title add.doi.url "title" output.check
1461    new.dblslash
1462    booktitle "booktitle" output.check
1463    new.slash
1464    editor.organization.after
1465    new.block
1466    output.address.publisher.date
1467    new.block
1468    format.bvolume output
1469    format.number.series output
1470    new.block
1471    format.pages.page output
1472    new.block
1473    note output
1474    new.block
1475    format.eprint output
1476    new.block
1477    output.url
1478    format.annote output
1479    fin.entry
1480 }
1481
```

```
1482 FUNCTION {inproceedings}
1483 {
1484    output.bibitem
1485    author.before
1486    new.sentence
1487    format.title add.doi.url "title" output.check
1488    new.dblslash
1489    booktitle "booktitle" output.check
1490    new.slash
1491    editor.organization.after
1492    new.block
```

```
1493    format.bvolume output
1494    format.number.series output
1495    new.block
1496    address empty$
1497      { publisher output
1498        format.date "year" output.check
1499      }
1500      { address output.nonnull
1501        new.colon
1502        publisher output
1503        format.date "year" output.check
1504      }
1505    if$
1506    new.block
1507    format.pages.page output
1508    new.block
1509    note output
1510    new.sentence
1511    format.eprint output
1512    new.block
1513    output.url
1514    format.annote output
1515    fin.entry
1516 }
1517
```

conference

```
1518 FUNCTION {conference} { inproceedings }
1519
```

manual

```
1520 FUNCTION {manual}
1521 {
1522    output.bibitem
1523    author empty$
1524      { organization empty$
1525          'skip$
1526          { organization output.nonnull
1527            address output
1528          }
1529        if$
1530      }
1531      { format.authors output.nonnull }
1532    if$
1533    new.block
1534    format.btitle "title" output.check
1535    author empty$
1536      { organization empty$
1537          { address new.block.checka
1538            address output
1539          }
1540          'skip$
1541        if$
1542      }
1543      { organization address new.block.checkb
1544        organization output
1545        address output
1546      }
1547    if$
1548    format.edition output
1549    format.date output
1550    new.block
1551    note output
1552    new.sentence
1553    format.eprint output
1554    new.block
1555    output.url
```

```
                1556   format.annote output
                1557   fin.entry
                1558 }
                1559

mastersthesis
                1560 FUNCTION {mastersthesis}
                1561 {
                1562   output.bibitem
                1563   format.authors "author" output.check
                1564   new.block
                1565   format.btitle add.doi.url "title" output.check
                1566   new.block
                1567   bbl.mthesis format.thesis.type output.nonnull
                1568   school "school" output.check
                1569   address output
                1570   format.date "year" output.check
                1571   new.block
                1572   note output
                1573   new.block
                1574   format.eprint output
                1575   new.block
                1576   output.url
                1577   format.annote output
                1578   fin.entry
                1579 }
                1580

       misc
                1581 FUNCTION {misc}
                1582 {
                1583   output.bibitem
                1584   format.authors "author" output.check
                1585   new.block
                1586   format.title "title" output.check
                1587   new.block
                1588   howpublished output
                1589   new.block
                1590   format.date output
                1591   new.block
                1592   note output
                1593   new.sentence
                1594   format.eprint output
                1595   new.block
                1596   output.url
                1597   format.annote output
                1598   fin.entry
                1599 }
                1600

    phdthesis
                1601 FUNCTION {phdthesis}
                1602 {
                1603   output.bibitem
                1604   format.authors "author" output.check
                1605   new.sentence
                1606   format.btitle add.doi.url "title" output.check
                1607   new.colon
                1608   bbl.phdthesis format.thesis.type output.nonnull
                1609   new.slash
                1610   school "school" output.check
                1611   new.block
                1612   output.address.publisher.date
                1613   new.block
                1614   format.pages.page output
                1615   note output
                1616   new.sentence
```

```
1617    format.eprint output
1618    new.block
1619    output.url
1620    format.annote output
1621    fin.entry
1622 }
1623
```

docdisser

```
1624 FUNCTION {docdisser}
1625 {
1626    output.bibitem
1627    format.authors "author" output.check
1628    new.sentence
1629    format.btitle add.doi.url "title" output.check
1630    new.colon
1631    bbl.docdisser format.thesis.type output.nonnull
1632    new.slash
1633    school "school" output.check
1634    new.block
1635    output.address.publisher.date
1636    new.block
1637    format.pages.page output
1638    note output
1639    new.sentence
1640    format.eprint output
1641    new.block
1642    output.url
1643    format.annote output
1644    fin.entry
1645 }
1646
```

proceedings

```
1647 FUNCTION {proceedings}
1648 {
1649    output.bibitem
1650    format.btitle add.doi.url "title" output.check
1651    new.slash
1652    editor.organization.after
1653    new.block
1654    format.bvolume output
1655    format.number.series output
1656    address empty$
1657      { publisher output
1658        format.date "year" output.check
1659      }
1660      { address output.nonnull
1661        format.date "year" output.check
1662        new.sentence
1663        publisher output
1664      }
1665    if$
1666    new.block
1667    note output
1668    new.sentence
1669    format.eprint output
1670    new.block
1671    output.url
1672    format.annote output
1673    fin.entry
1674 }
1675
```

techreport

```
1676 FUNCTION {techreport}
1677 {
```

```
1678    output.bibitem
1679    author.before
1680    new.sentence
1681    format.title add.doi.url "title" output.check
1682    new.colon
1683    format.tr.number output.nonnull
1684    editor.organization.after
1685    new.block
1686    address output
1687    new.colon
1688    institution "institution" output.check
1689    format.date "year" output.check
1690    new.block
1691    note output
1692    new.sentence
1693    format.eprint output
1694    new.block
1695    output.url
1696    format.annote output
1697    fin.entry
1698 }
1699
```

patent

```
1700 FUNCTION {patent}
1701 {
1702    output.bibitem
1703    title output.nonnull
1704    add.blank
1705    media in.square.brackets output
1706    new.colon
1707    format.type.number output
1708    add.blank
1709    location output
1710    new.colon
1711    ipc output
1712    new.slash
1713    format.authors.after "author" output.check
1714    add.blank
1715    authortype in.round.brackets output.nonnull
1716    new.semicolon
1717    holder output.nonnull
1718    new.semicolon
1719    credits output.nonnull
1720    new.dash
1721    reqnumber add.number output
1722    new.semicolon
1723    format.reqdate output
1724    new.semicolon
1725    format.pubdate output
1726    publication output
1727    new.semicolon
1728    format.prdate output
1729    prnumber output
1730    prcountry in.round.brackets output
1731    new.block
1732    note output
1733    new.sentence
1734    output.url
1735    format.annote output
1736    new.block
1737    pagetotal output
1738    fin.entry
1739 }
1740
```

unpublished

```
1741 FUNCTION {unpublished}
1742 {
1743   output.bibitem
1744   author.before
1745   format.btitle add.doi.url "title" output.check
1746   editor.organization.after
1747   new.block
1748   note "note" output.check
1749   new.sentence
1750   format.eprint output
1751   new.block
1752   output.url
1753   format.annote output
1754   fin.entry
1755 }
1756
```

default.type

```
1757 FUNCTION {default.type} { misc }
1758
```

1.10 Сокращения названий месяцев

```
1759 MACRO {jan} {"\bbljan{}"}
1760 MACRO {feb} {"\bblfeb{}"}
1761 MACRO {mar} {"\bblmar{}"}
1762 MACRO {apr} {"\bblapr{}"}
1763 MACRO {may} {"\bblmay{}"}
1764 MACRO {jun} {"\bbljun{}"}
1765 MACRO {jul} {"\bbljul{}"}
1766 MACRO {aug} {"\bblaug{}"}
1767 MACRO {sep} {"\bblsep{}"}
1768 MACRO {oct} {"\bbloct{}"}
1769 MACRO {nov} {"\bblnov{}"}
1770 MACRO {dec} {"\bbldec{}"}
1771
```

1.11 Сокращения названий журналов

1.11.1 Физика и астрономия

Код данной секции создан на основе файла physjour.mbs пакета custom-bib.

```
1772 MACRO {aa}{"Astron. \& Astrophys."}
1773 MACRO {aasup}{"Astron. \& Astrophys. Suppl. Ser."}
1774 MACRO {aj} {"Astron. J."}
1775 MACRO {aph} {"Acta Phys."}
1776 MACRO {advp} {"Adv. Phys."}
1777 MACRO {ajp} {"Amer. J. Phys."}
1778 MACRO {ajm} {"Amer. J. Math."}
1779 MACRO {amsci} {"Amer. Sci."}
1780 MACRO {anofd} {"Ann. Fluid Dyn."}
1781 MACRO {am} {"Ann. Math."}
1782 MACRO {ap} {"Ann. Phys. (NY)"}
1783 MACRO {adp} {"Ann. Phys. (Leipzig)"}
1784 MACRO {ao} {"Appl. Opt."}
1785 MACRO {apl} {"Appl. Phys. Lett."}
1786 MACRO {app} {"Astroparticle Phys."}
1787 MACRO {apj} {"Astrophys. J."}
1788 MACRO {apjsup} {"Astrophys. J. Suppl."}
1789 MACRO {apss} {"Astrophys. Space Sci."}
1790 MACRO {araa} {"Ann. Rev. Astron. Astrophys."}
1791 MACRO {baas} {"Bull. Amer. Astron. Soc."}
1792 MACRO {baps} {"Bull. Amer. Phys. Soc."}
1793 MACRO {cmp} {"Comm. Math. Phys."}
1794 MACRO {cpam} {"Commun. Pure Appl. Math."}
1795 MACRO {cppcf} {"Comm. Plasma Phys. \& Controlled Fusion"}
1796 MACRO {cpc} {"Comp. Phys. Comm."}
```

```
1797 MACRO {cqg} {"Class. Quant. Grav."}
1798 MACRO {cra} {"C. R. Acad. Sci. A"}
1799 MACRO {fed} {"Fusion Eng. \& Design"}
1800 MACRO {ft} {"Fusion Tech."}
1801 MACRO {grg} {"Gen. Relativ. Gravit."}
1802 MACRO {ieeens} {"IEEE Trans. Nucl. Sci."}
1803 MACRO {ieeeps} {"IEEE Trans. Plasma Sci."}
1804 MACRO {ijimw} {"Interntl. J. Infrared \& Millimeter Waves"}
1805 MACRO {ip} {"Infrared Phys."}
1806 MACRO {irp} {"Infrared Phys."}
1807 MACRO {jap} {"J. Appl. Phys."}
1808 MACRO {jasa} {"J. Acoust. Soc. America"}
1809 MACRO {jcp} {"J. Comp. Phys."}
1810 MACRO {jchp} {"J. Chem. Phys."}
1811 MACRO {jetp} {"Sov. Phys.--JETP"}
1812 MACRO {jfe} {"J. Fusion Energy"}
1813 MACRO {jfm} {"J. Fluid Mech."}
1814 MACRO {jmp} {"J. Math. Phys."}
1815 MACRO {jne} {"J. Nucl. Energy"}
1816 MACRO {jnec} {"J. Nucl. Energy, C: Plasma Phys., Accelerators, Thermonucl. Res."}
1817 MACRO {jnm} {"J. Nucl. Mat."}
1818 MACRO {jpc} {"J. Phys. Chem."}
1819 MACRO {jpp} {"J. Plasma Phys."}
1820 MACRO {jpsj} {"J. Phys. Soc. Japan"}
1821 MACRO {jsi} {"J. Sci. Instrum."}
1822 MACRO {jvst} {"J. Vac. Sci. \& Tech."}
1823 MACRO {nat} {"Nature"}
1824 MACRO {nature} {"Nature"}
1825 MACRO {nedf} {"Nucl. Eng. \& Design/Fusion"}
1826 MACRO {nf} {"Nucl. Fusion"}
1827 MACRO {nim} {"Nucl. Inst. \& Meth."}
1828 MACRO {nimpr} {"Nucl. Inst. \& Meth. in Phys. Res."}
1829 MACRO {np} {"Nucl. Phys."}
1830 MACRO {npb} {"Nucl. Phys. B"}
1831 MACRO {nt/f} {"Nucl. Tech./Fusion"}
1832 MACRO {npbpc} {"Nucl. Phys. B (Proc. Suppl.)"}
1833 MACRO {inc} {"Nuovo Cimento"}
1834 MACRO {nc} {"Nuovo Cimento"}
1835 MACRO {pf} {"Phys. Fluids"}
1836 MACRO {pfa} {"Phys. Fluids A: Fluid Dyn."}
1837 MACRO {pfb} {"Phys. Fluids B: Plasma Phys."}
1838 MACRO {pl} {"Phys. Lett."}
1839 MACRO {pla} {"Phys. Lett. A"}
1840 MACRO {plb} {"Phys. Lett. B"}
1841 MACRO {prep} {"Phys. Rep."}
1842 MACRO {pnas} {"Proc. Nat. Acad. Sci. USA"}
1843 MACRO {pp} {"Phys. Plasmas"}
1844 MACRO {ppcf} {"Plasma Phys. \& Controlled Fusion"}
1845 MACRO {phitrsl} {"Philos. Trans. Roy. Soc. London"}
1846 MACRO {prl} {"Phys. Rev. Lett."}
1847 MACRO {pr} {"Phys. Rev."}
1848 MACRO {physrev} {"Phys. Rev."}
1849 MACRO {pra} {"Phys. Rev. A"}
1850 MACRO {prb} {"Phys. Rev. B"}
1851 MACRO {prc} {"Phys. Rev. C"}
1852 MACRO {prd} {"Phys. Rev. D"}
1853 MACRO {pre} {"Phys. Rev. E"}
1854 MACRO {ps} {"Phys. Scripta"}
1855 MACRO {procrsl} {"Proc. Roy. Soc. London"}
1856 MACRO {rmp} {"Rev. Mod. Phys."}
1857 MACRO {rsi} {"Rev. Sci. Inst."}
1858 MACRO {science} {"Science"}
1859 MACRO {sciam} {"Sci. Am."}
1860 MACRO {sam} {"Stud. Appl. Math."}
1861 MACRO {sjpp} {"Sov. J. Plasma Phys."}
1862 MACRO {spd} {"Sov. Phys.--Doklady"}
```

```
1863 MACRO {sptp} {"Sov. Phys.--Tech. Phys."}
1864 MACRO {spu} {"Sov. Phys.--Uspekhi"}
1865 MACRO {st} {"Sky and Telesc."}
1866
```

1.11.2 Оптика

Из файла `photjour.mbs`.

```
1867 MACRO {appopt} {"Appl. Opt."}
1868 MACRO {bell} {"Bell Syst. Tech. J."}
1869 MACRO {ell} {"Electron. Lett."}
1870 MACRO {jasp} {"J. Appl. Spectr."}
1871 MACRO {jqe} {"IEEE J. Quantum Electron."}
1872 MACRO {jlwt} {"J. Lightwave Technol."}
1873 MACRO {jmo} {"J. Mod. Opt."}
1874 MACRO {josa} {"J. Opt. Soc. America"}
1875 MACRO {josaa} {"J. Opt. Soc. Amer.~A"}
1876 MACRO {josab} {"J. Opt. Soc. Amer.~B"}
1877 MACRO {jdp} {"J. Phys. (Paris)"}
1878 MACRO {oc} {"Opt. Commun."}
1879 MACRO {ol} {"Opt. Lett."}
1880 MACRO {os} {"Opt. Spectrosc."}
1881 MACRO {phtl} {"IEEE Photon. Technol. Lett."}
1882 MACRO {pspie} {"Proc. Soc. Photo-Opt. Instrum. Eng."}
1883 MACRO {sjot} {"Sov. J. Opt. Technol."}
1884 MACRO {sjqe} {"Sov. J. Quantum Electron."}
1885 MACRO {sleb} {"Sov. Phys.--Leb. Inst. Rep."}
1886 MACRO {stph} {"Sov. Phys.--Techn. Phys."}
1887 MACRO {stphl} {"Sov. Techn. Phys. Lett."}
1888 MACRO {vr} {"Vision Res."}
1889 MACRO {zph} {"Z. f. Physik"}
1890 MACRO {zphb} {"Z. f. Physik~B"}
1891 MACRO {zphd} {"Z. f. Physik~D"}
1892
```

1.11.3 Физика конденсированного состояния

```
1893 MACRO {sse} {"Solid-State Electron."}
1894 MACRO {pss} {"Phys. Sol. State"}
1895 MACRO {sst} {"Semicond. Sci. Tech."}
1896 MACRO {nl} {"Nano Lett."}
1897
1898 READ
1899
```

1.12   Сортировка списка литературы по именам авторов и названиям

```
1900 ⟨*gost705s⟩
```

sortify
```
1901 FUNCTION {sortify}
1902 {
1903   purify$
1904   "l" change.case$
1905 }
1906
1907 INTEGERS { len }
1908
```

chop.word
```
1909 FUNCTION {chop.word}
1910 {
1911   's :=
1912   'len :=
1913   s #1 len substring$ =
1914     { s len #1 + global.max$ substring$ }
1915     's
1916   if$
```

```
                        1917 }
                        1918

sort.format.names
                        1919 FUNCTION {sort.format.names}
                        1920 {
                        1921   's :=
                        1922   #1 'nameptr :=
                        1923   ""
                        1924   s num.names$ 'numnames :=
                        1925   numnames 'namesleft :=
                        1926     { namesleft #0 > }
                        1927     { nameptr #1 >
                        1928         { "  " * }
                        1929         'skip$
                        1930       if$
                        1931       s nameptr
                        1932       "{vv{ } }{ll{ }}{  f{ }}{  jj{ }}"
                        1933       format.name$ 't :=
                        1934       nameptr numnames = t "others" = and
                        1935         { "et al" * }
                        1936         { t sortify * }
                        1937       if$
                        1938       nameptr #1 + 'nameptr :=
                        1939       namesleft #1 - 'namesleft :=
                        1940     }
                        1941   while$
                        1942 }
                        1943

sort.format.title
                        1944 FUNCTION {sort.format.title}
                        1945 {
                        1946   't :=
                        1947   "A " #2
                        1948     "An " #3
                        1949       "The " #4 t chop.word
                        1950     chop.word
                        1951   chop.word
                        1952   sortify
                        1953   #1 global.max$ substring$
                        1954 }
                        1955

author.sort
                        1956 FUNCTION {author.sort}
                        1957 {
                        1958   author empty$
                        1959     { key empty$
                        1960         { "to sort, need author or key in " cite$ * warning$ "" }
                        1961         { key sortify }
                        1962       if$
                        1963     }
                        1964     { author sort.format.names }
                        1965   if$
                        1966 }
                        1967

author.title.sort
                        1968 FUNCTION {author.title.sort}
                        1969 {
                        1970   author empty$
                        1971     { title empty$
                        1972       { key empty$
                        1973         { "to sort, need author, title, or key in " cite$ * warning$ "" }
                        1974         { key sortify }
                        1975       if$
                        1976       }
                        1977       { title sort.format.title }
```

34

```
1978    if$
1979    }
1980    { author sort.format.names }
1981  if$
1982 }
1983
```

presort

```
1984 FUNCTION {presort}
1985 {
1986   author.title.sort
1987   "    "
1988   *
1989   year field.or.null sortify
1990   *
1991   "    "
1992   *
1993   title field.or.null
1994   sort.format.title
1995   *
1996   #1 entry.max$ substring$
1997   'sort.key$ :=
1998 }
1999
2000 ITERATE {presort}
2001
2002 SORT
2003
2004 ⟨/gost705s⟩
2005
```

1.13 Вычисление длины табуляции между номером и текстом

```
2006 STRINGS { longest.label }
2007
2008 INTEGERS { number.label longest.label.width }
2009
```

initialize.longest.label

```
2010 FUNCTION {initialize.longest.label}
2011 {
2012   "" 'longest.label :=
2013   #1 'number.label :=
2014   #0 'longest.label.width :=
2015 }
2016
```

longest.label.pass

```
2017 FUNCTION {longest.label.pass}
2018 {
2019   number.label int.to.str$ 'label :=
2020   number.label #1 + 'number.label :=
2021   label width$ longest.label.width >
2022     { label 'longest.label :=
2023       label width$ 'longest.label.width :=
2024     }
2025   'skip$
2026   if$
2027 }
2028
2029 EXECUTE {initialize.longest.label}
2030
2031 ITERATE {longest.label.pass}
2032
```

1.14 Формирование окружения thebibliography

1.14.1 Преамбула

begin.bib

```
2033 FUNCTION {begin.bib}
2034 {
2035   preamble$ empty$
2036     'skip$
2037     { preamble$ write$ newline$ }
2038   if$
2039   "\ifx\undefined\BibEmph\def\BibEmph#1{#1}\else\fi"
2040   write$ newline$
2041   "\ifx\undefined\href\def\href#1#2{#2}\else\fi"
2042   write$ newline$
2043   "\ifx\undefined\url\def\url#1{\texttt{#1}}\else\fi"
2044   write$ newline$
2045   "\ifx\undefined\urlprefix\def\urlprefix{URL: }\else\fi"
2046   write$ newline$
2047   "\ifx\undefined\BibUrl\def\BibUrl#1{\urlprefix\url{#1}}\else\fi"
2048   write$ newline$
2049   "\ifx\undefined\BibUrlDate\long\def\BibUrlDate#1{({%"
2050   write$ newline$
2051   "\cyr\cyrd\cyra\cyrt\cyra\ %"
2052   write$ newline$
2053   "\cyro\cyrb\cyrr\cyra\cyrshch\cyre\cyrn\cyri\cyrya: #1)}\else\fi"
2054   write$ newline$
2055   "\ifx\undefined\BibAnnote\long\def\BibAnnote#1{#1}\else\fi"
2056   write$ newline$
2057   "\begin{thebibliography}{" longest.label  * "}" *
2058   write$ newline$
2059   "\def\selectlanguageifdefined#1{"
2060   write$ newline$
2061   "\expandafter\ifx\csname date#1\endcsname\relax"
2062   write$ newline$
2063   "\else\language\csname l@#1\endcsname\fi}"
2064   write$ newline$
2065 }
2066
```

1.14.2   Начало окружения

```
2067 EXECUTE {begin.bib}
2068
2069 EXECUTE {init.state.consts}
2070
2071 ITERATE {call.type$}
2072
```

1.14.3   Конец окружения

end.bib

```
2073 FUNCTION {end.bib}
2074 {
2075   newline$
2076   "\end{thebibliography}" write$ newline$
2077 }
2078
2079 EXECUTE {end.bib}
2080
2081 ⟨/gost705 | gost705s⟩
```

2 Предметный указатель

Функции