INTEGERS uppercase.name max.num.authors period.between.author.year sentence.case.title link.title show.mark show.medium.type slash.for.extraction in.booktitle italic.jounal bold.journal.volume show.missing.address.publ show.url show.doi show.note

FUNCTION load.config 0 'uppercase.name := 3 'max.num.authors := 1 'sentence.case.title := 0 'link.title := 1 'show.mark := 1 'show.medium.type := 1 'slash.for.extraction := 0 'in.booktitle := 0 'italic.jounal := 0 'bold.journal.volume := 1 'show.missing.address.publisher := 1 'show.url := 1 'show.doi := 0 'show.note :=

ENTRY address author booktitle date doi edition editor howpublished institution journal key language mark medium note number organization pages publisher school series title translator url urldate volume year entry.lang entry.is.electronic entry.numbered label extra.label sort.label short.list entry.mark entry.url

INTEGERS output.state before.all mid.sentence after.sentence after.block after.slash

INTEGERS lang.zh lang.ja lang.en lang.ru lang.other

INTEGERS charptr len

FUNCTION init.state.consts 0 'before.all := 1 'mid.sentence := 2 'after.sentence := 3 'after.block := 4 'after.slash := 3 'lang.zh := 4 'lang.ja := 1 'lang.en := 2 'lang.ru := 0 'lang.other :=

FUNCTION bbl.anonymous entry.lang lang.zh = "" "Anon" if

FUNCTION bbl.space entry.lang lang.zh = " " " " if

FUNCTION bbl.et.al entry.lang lang.zh = "" entry.lang lang.ja = "" entry.lang lang.ru = "" "et al." if*if* if

FUNCTION citation.et.al bbl.et.al

FUNCTION bbl.colon ": "

FUNCTION bbl.wide.space " "

FUNCTION bbl.slash "//"

FUNCTION bbl.sine.loco entry.lang lang.zh = "[]" "[S.l.]" if

FUNCTION bbl.sine.nomine entry.lang lang.zh = "[]" "[s.n.]" if

FUNCTION bbl.sine.loco.sine.nomine entry.lang lang.zh = "[:]" "[S.l.: s.n.]" if

FUNCTION not 0 1 if

FUNCTION and 'skip*pop* 0 if

FUNCTION or pop1′*skip* if

STRINGS s t

FUNCTION output.nonnull 's := output.state mid.sentence = ", " * write*output.stateafter.block = add.period* write*newline* "" write*output.statebefore.all =′ write* output.state after.slash = bbl.slash * write*newline* add.period"" ∗ *write* if*if* if*mid.sentence′output.state* := *if* s

FUNCTION output duplicate*empty* 'pop′*output.nonnullif*

FUNCTION output.after 't := duplicate*empty* 'pop′*s* := *output.statemid.sentence* = *t* ∗ *write* output.state after.block = add.period*write* newline""*write* output.state before.all = 'write*output.stateafter.slash* = *bbl.sla* add.period"" ∗ *write* if*if* if*mid.sentence′output.state* := *if* s if

FUNCTION output.check 't := duplicate*empty* pop"*empty*"*t*∗"*in*" ∗*cite* * warning′*output.nonnullif*

FUNCTION fin.entry add.period*write* newline

FUNCTION new.block output.state before.all = 'skip*output.stateafter.slash =′ skip* after.block 'output.state := if*if*

FUNCTION new.sentence output.state after.block = 'skip*output.statebefore.all =′ skip* output.state after.slash = 'skip*after.sentence′output.state* :=*if* if*if*

FUNCTION new.slash output.state before.all = 'skip*slash.for.extractionafter.slash′output.state* :=*after.block′out* if

FUNCTION new.block.checka empty′*skip* 'new.block if

FUNCTION new.block.checkb empty*swap* empty*and′skip* 'new.block if

FUNCTION new.sentence.checka empty′*skip* 'new.sentence if

FUNCTION new.sentence.checkb empty*swap* empty*and′skip* 'new.sentence if

FUNCTION field.or.null duplicate*empty* pop""′*skip* if

FUNCTION italicize duplicate*empty* pop""""*swap* * "" * if

INTEGERS byte second.byte

INTEGERS char.lang tmp.lang

STRINGS tmp.str

FUNCTION get.str.lang 'tmp.str := lang.other 'tmp.lang := 1 'charptr := tmp.str text.length1+′*len* := *charptrlen* <*tmp.strcharptr*1*substring* chr.to.int′*byte* := *byte*128 < *charptr*1 +′ *charptr* := *byte*64 > *byte*91 < *andbyte*96 tmp.str charptr 1 + 1 substring*chr.to.int* 'second.byte := byte 224 ¡ charptr 2 + 'charptr := byte 207 ¿ byte 212 ¡ and byte 212 = second.byte 176 ¡ and or lang.ru 'char.lang := lang.other 'char.lang := if*byte*240 < *charptr*3 +′ *charptr* := *byte*227 > *byte*234 < *andlang.zh′char.lang* :=*byte*227 = *second.byte*143 > *lang.zh′ch* if*byte*239 = *second.byte*163 > *second.byte*172 < *andandlang.zh′char.lang* :=*lang.other′char.lang* :=*if* if*if* charptr 4 + 'charptr := byte 240 = second.byte 159 ¿ and lang.zh 'char.lang := lang.other 'char.lang := if*if* if*if* char.lang tmp.lang ¿ char.lang 'tmp.lang := 'skip*if* while*tmp.lang*

FUNCTION check.entry.lang author field.or.null title field.or.null * get.str.lang

FUNCTION set.entry.lang language empty*check.entry.langlanguage*"*english*" = *language*"*american*" = *orlangua* if*if* if*if* 'entry.lang :=

FUNCTION set.entry.numbered type"*patent*" = *type* "standard" = or type"*techreport*" = *or*1′*entry.numbered* :=0′

INTEGERS nameptr namesleft numnames name.lang

FUNCTION format.names 's := 1 'nameptr := s num.names′*numnames* := *numnames′namesleft* := *namesleft*0 >*snameptr*"*vv ll*, *jj*, *ff*" *format.name* 't := nameptr max.num.authors ¿ bbl.et.al 1 'namesleft := t "others" = bbl.et.al t get.str.lang 'name.lang := name.lang lang.en = t 1 "vv ll f" format.name*uppercase.name*"*u*" 'skip*if* t 1 ", jj" format.name∗*t*1"*llff*" *format.name* if*if* if*nameptr*1 > ","*swap* * * 'skip*if* nameptr 1 + 'nameptr := namesleft 1 - 'namesleft := while

FUNCTION format.key empty*keyfield.or.null*""*if*

FUNCTION format.authors author empty*notauthorformat.names*"*emptyauthorin*"*cite* * warning""*if*

FUNCTION format.editors editor empty""*editorformat.namesif*

FUNCTION format.translators translator empty""*translatorformat.namesentry.langlang.zh* = *translatornum.n* 3 ¿ "" * ", " * if′*skip* if*if*

FUNCTION format.full.names 's := 1 'nameptr := s num.names′*numnames* := *numnames′namesleft* := *namesleft*0 >*snameptr*"*vv ll*, *jj*, *ff*" *format.name* 't := t get.str.lang 'name.lang := name.lang lang.en = t 1 "vv ll" format.name′*t* := *t*1"*llff*" *format.name* 't := if*nameptr*1 > *namesleft*1 > "," ∗ *t*∗*numnames*2 > ","∗′*skip* if*t*"*others*" = "*et al.*"∗"*and*" ∗ *t*∗*if* if′*tif* nameptr 1 + 'nameptr := namesleft 1 - 'namesleft := while

FUNCTION author.editor.full author empty*editorempty* "" editor format.full.names if*authorformat.full.namesi*

FUNCTION author.full author empty""*authorformat.full.namesif*

FUNCTION editor.full editor empty""*editorformat.full.namesif*

FUNCTION make.full.names type"*book*" = *type* "inbook" = or 'author.editor.full type"*collection*" = *type* "proceedings" = or 'editor.full 'author.full if*if*

FUNCTION output.bibitem newline"["*write* label ")" * make.full.names duplicate*short.list* = *pop* * if′*s* := *stext.length* 'charptr := charptr 0 ¿ s charptr 1 substring"[" = *notandcharptr*1 −′ *charptr* :=*while* charptr 0 ¿ "" s * "" * s if"]" ∗ *write* cite*write* "" write*newline* "" before.all 'output.state :=

FUNCTION change.sentence.case entry.lang lang.en = "t" change.case′*skip* if

FUNCTION add.link url empty*not*""*url* ∗ "" ∗ *swap* * "" * doi empty*not*"*http* : //*dx.doi.org*/"*doi* ∗ "" ∗ *swap* * "" * 'skip*if* if

FUNCTION format.title title empty""*titlesentence.case.title′change.sentence.case′skip* if*entry.numberednumbere* not and bbl.colon * number * 'skip*if* link.title 'add.link 'skip*if* if

FUNCTION tie.or.space.connect duplicate*text.length* 3 ¡ " " " " if*swap* * *

FUNCTION either.or.check empty′*pop* "can't use both " swap∗"*fieldsin*" ∗ *cite* * warning*if*

FUNCTION is.digit duplicate*empty* pop0*chr.to.int* duplicate"0"*chr.to.int* ¡ pop0"9"*chr.to.int* ¿ 0 1 if*if* if

FUNCTION is.number 's := s empty0*stext.length* 'charptr := charptr 0 ¿ s charptr 1 substring*is.digitandcharptr*1 − charptr not if

FUNCTION format.volume volume empty*notvolumeis.numberentry.langlang.zh* = ""*volume* ∗ ""∗"*volume*"*volum* volume if""*if*

FUNCTION format.number number empty*notnumberis.numberentry.langlang.zh* = ""*number* ∗ ""∗"*number*"*num*
number if""*if*

FUNCTION format.volume.number volume empty*notformat.volumeformat.numberif*

FUNCTION format.title.vol.num title sentence.case.title 'change.sentence.case 'skip*if* entry.numbered
number empty*notbbl.colon* ∗ *number*∗′*skip* if*format.volume.number′s* := *sempty* not bbl.colon * s * 'skip*if*
if

FUNCTION format.series.vol.num.title format.volume.number 's := series empty*notseriessentence.case.title′change*
if*entry.numberedbbl.wide.space*∗*bbl.colon* ∗ *sempty* not s * bbl.wide.space * 'skip*if* if*title*∗*sentence.case.title′change.se*
if*entry.numberednumberempty* not and bbl.colon * number * 'skip*if* format.title.vol.num if*link.title′add.link′skip*
if

FUNCTION format.booktitle.vol.num booktitle entry.numbered 'skip*format.volume.number′s* := *sempty*
not bbl.colon * s * 'skip*if* if

FUNCTION format.series.vol.num.booktitle format.volume.number 's := series empty*notseriesbbl.colon* ∗ *entry.num*
not and s * bbl.wide.space * 'skip*if* booktitle * format.booktitle.vol.num if*in.booktitleduplicate* empty*notentry.langlan*
and"*In* : "*swap* * 'skip*if* 'skip*if*

FUNCTION format.journal journal italic.jounal entry.lang lang.en = and 'italicize 'skip*if*

FUNCTION set.entry.mark entry.mark empty*not′pop* mark empty*notpop* mark 'entry.mark := 'en-
try.mark := if*if*

FUNCTION format.mark show.mark entry.mark show.medium.type medium empty*not*"/" ∗ *medium*∗*entry.is.elect*
if*if* 'skip*if* 'entry.mark := "[" entry.mark * "]" * "" if

FUNCTION num.to.ordinal duplicate*text.length* 'charptr := duplicate*charptr*1*substring* 's := s "1"
= "st" * s "2" = "nd" * s "3" = "rd" * "th" * if*if* if

FUNCTION format.edition edition empty""*editionis.numberentry.langlang.zh* = *edition*""∗*editionnum.to.ordinal*
entry.lang lang.en = edition change.sentence.case 's := s "Revised" = s "Revised edition" = or "Rev. ed."
s " ed." * if*editionif* if*if*

FUNCTION format.publisher publisher empty*notpublisherschoolempty* not school organization
empty*notorganizationinstitutionempty* not institution "" if*if* if*if*

FUNCTION format.address.publisher address empty*notaddressformat.publisherempty* not bbl.colon
* format.publisher * entry.is.electronic not show.missing.address.publisher and bbl.colon * bbl.sine.nomine *
'skip*if* if*entry.is.electronicnotshow.missing.address.publisherandformat.publisherempty* not bbl.sine.loco
bbl.colon * format.publisher * bbl.sine.loco.sine.nomine if*format.publisherempty* not format.publisher
"" if*if* if

FUNCTION extract.before.dash duplicate*empty* pop""*′s* := 1*′charptr* := *stext.length* 1 + 'len :=
charptr len ¡ s charptr 1 substring" − " = *notandcharptr*1 +′ *charptr* :=*while* s 1 charptr 1 - substring*if*

FUNCTION extract.after.dash duplicate*empty* pop""*′s* := 1*′charptr* := *stext.length* 1 + 'len := charptr
len ¡ s charptr 1 substring"−" = *notandcharptr*1 +′ *charptr* :=*while* charptr len ¡ s charptr 1 substring"−" =
*andcharptr*1 +′ *charptr* :=*while* s charptr global.max*substring* if

FUNCTION contains.dash duplicate*empty* pop0′*s* := *sempty* not s 1 1 substring"−" = *notands*2*global.max*
substring′*s* := *while* s empty*notif*

FUNCTION format.year year empty*notyearextract.before.dashdateempty* not date extract.before.dash
"empty year in " cite∗*warning* urldate empty*not*"["*urldateextract.before.dash* ∗ "]"∗""*if* if*if* extra.label
*

FUNCTION format.date type"*patent*" = *type* "newspaper" = or date empty*notanddateyearif*

FUNCTION format.editdate date empty*not*"("*date* ∗ ")"∗""*if*

FUNCTION format.urldate urldate empty*notentry.is.electronicand*"["*urldate* ∗ "]"∗""*if*

FUNCTION hyphenate 't := "" t empty*nott*11*substring* "-" = "-" * t 1 1 substring" − " =
*t*2*global.max* substring′*t* := *while* t 1 1 substring∗*t*2*global.max* substring′*t* := *if* while

FUNCTION format.pages pages empty""*pageshyphenateif*

FUNCTION format.journal.volume volume empty*notbold.journal.volume*""*volume* ∗ ""∗*volumeif* ""
if

FUNCTION format.journal.number number empty*not*"("*number* ∗ ")"∗""*if*

FUNCTION format.journal.pages pages empty""" : "*pageshyphenate*∗*if*

FUNCTION format.periodical.year.volume.number year empty$notyearextract.before.dash" emptyyearinperiodical$
* warningif volume empty$not", " * volumeextract.before.dash *' skip$ if$numberempty$ not "(" * number extract.before.dash * ")" * 'skipif year contains.dash "–" * year extract.after.dash empty$volumeextract.after.dashempty$
and number extract.after.dash empty$andnotyearextract.after.dashempty$ not year extract.after.dash *
year extract.before.dash * if$volumeempty$ not ", " * volume extract.after.dash * 'skipif number empty$not"(" * numberex$
if$'skip$ if$'skip$ if

FUNCTION check.url url empty$not""url * "" *' entry.url := 1'entry.is.electronic :=howpublishedempty$
not howpublished 1 5 substring"" $= howpublished'entry.url := 1'entry.is.electronic :=' skip$ if$noteempty$
not note 1 5 substring"" $= note'entry.url := 1'entry.is.electronic :=' skip$ if$'skip$ ifif if

FUNCTION format.url entry.url empty$notnew.blockentry.url""if$

FUNCTION check.doi doi empty$not1'entry.is.electronic :=' skip$ if

FUNCTION is.in.url 's := s empty$1entry.urlempty$ 0 s text.length$'len := entry.urltext.length$
'charptr := entry.url charptr len substring$s = notcharptr0 > andcharptr1 -' charptr :=while$ charptr ifif

FUNCTION format.doi "" doi empty$notshow.doiand""'s := doi't := 0'numnames := tempty$ not t 1 1
substring$'tmp.str := tmp.str", " = tmp.str"" = ort21substring$ empty$ort21substring$ empty$stmp.str *' s :=' skip$
if$sempty$ s is.in.url or 'skip$numnames1 +' numnames := numnames1 > ", " * "DOI : " * if$ "" s * "" * *
if$""'s := stmp.str *' s :=if$ t 2 global.max$substring$ 't := while$'s := sempty$ not new.block s "" if$'skip$ if

FUNCTION check.electronic "" 'entry.url := 0 'entry.is.electronic := 'check.doi 'skipif 'check.url 'skipif
medium empty$notmedium"MT" = medium"DK" = ormedium"CD" = ormedium"OL" = or1'entry.is.electronic :=' s$
if$'skip$ if

FUNCTION format.note note empty$notshow.noteandnote""if$

FUNCTION empty.misc.check author empty$titleempty$ year empty$andandkeyempty$ not and "all relevant fields are empty in " cite$*warning$ 'skipif

FUNCTION monograph output.bibitem author empty$notformat.authorseditorempty$ not format.editors
"empty author and editor in " cite$*warning$ "" ifif output new.block format.series.vol.num.title "title" output.check "M" set.entry.mark format.mark "" output.after new.block format.translators output new.sentence format.edition output new.block format.publisher output format.year "year" output.check format.pages ". " output.after format.urldate "" output.after format.url output format.doi output new.block format.note output fin.entry

FUNCTION incollection output.bibitem format.authors output author format.key output new.block format.title "title" output.check "M" set.entry.mark format.mark "" output.after new.block format.translators output new.block format.editors output new.block format.series.vol.num.booktitle "booktitle" output.check new.block format.edition output new.block format.publisher output format.year "year" output.check format.pages ". " output.after format.urldate "" output.after format.url output format.doi output new.block format.note output fin.entry

FUNCTION periodical output.bibitem format.authors output author format.key output new.block format.title "title" output.check "J" set.entry.mark format.mark "" output.after new.block format.periodical.year.volume.nu
output new.block format.address.publisher output format.date "year" output.check format.urldate "" output.after format.url output format.doi output new.block format.note output fin.entry

FUNCTION article output.bibitem format.authors output author format.key output new.block format.title "title" output.check "J" set.entry.mark format.mark "" output.after new.block format.journal "journal" output.check format.date "year" output.check format.journal.volume output format.journal.number
"" output.after format.journal.pages "" output.after format.urldate "" output.after format.url output format.doi output new.block format.note output fin.entry

FUNCTION patent output.bibitem format.authors output author format.key output new.block format.title "title" output.check "P" set.entry.mark format.mark "" output.after new.block format.date "year" output.check format.urldate "" output.after format.url output format.doi output new.block format.note output fin.entry

FUNCTION electronic 1 1 check.electronic 1 'entry.is.electronic := output.bibitem format.authors output author format.key output new.block format.series.vol.num.title "title" output.check "EB" set.entry.mark format.mark "" output.after new.block format.address.publisher output format.pages bbl.colon output.after format.editdate "" output.after format.urldate "" output.after format.url output format.doi output date empty$format.dateoutput'skip$ if$new.blockformat.noteoutputfin.entry$

FUNCTION misc journal emptynot'articlebooktitleempty not 'incollection publisher emptynot'monographentry.is.
if$ if$empty.misc.check
FUNCTION archive "A" set.entry.mark misc
FUNCTION book monograph
FUNCTION booklet book
FUNCTION collection "G" set.entry.mark monograph
FUNCTION database "DB" set.entry.mark electronic
FUNCTION dataset "DS" set.entry.mark electronic
FUNCTION inbook book
FUNCTION inproceedings "C" set.entry.mark incollection
FUNCTION conference inproceedings
FUNCTION map "CM" set.entry.mark misc
FUNCTION manual monograph
FUNCTION mastersthesis "D" set.entry.mark monograph
FUNCTION newspaper "N" set.entry.mark article
FUNCTION online "EB" set.entry.mark electronic
FUNCTION phdthesis mastersthesis
FUNCTION proceedings "C" set.entry.mark monograph
FUNCTION software "CP" set.entry.mark electronic
FUNCTION standard "S" set.entry.mark misc
FUNCTION techreport "R" set.entry.mark misc
FUNCTION unpublished "Z" set.entry.mark misc
FUNCTION default.type misc
MACRO jan "January"
MACRO feb "February"
MACRO mar "March"
MACRO apr "April"
MACRO may "May"
MACRO jun "June"
MACRO jul "July"
MACRO aug "August"
MACRO sep "September"
MACRO oct "October"
MACRO nov "November"
MACRO dec "December"
MACRO acmcs "ACM Computing Surveys"
MACRO acta "Acta Informatica"
MACRO cacm "Communications of the ACM"
MACRO ibmjrd "IBM Journal of Research and Development"
MACRO ibmsj "IBM Systems Journal"
MACRO ieeese "IEEE Transactions on Software Engineering"
MACRO ieeetc "IEEE Transactions on Computers"
MACRO ieeetcad "IEEE Transactions on Computer-Aided Design of Integrated Circuits"
MACRO ipl "Information Processing Letters"
MACRO jacm "Journal of the ACM"
MACRO jcss "Journal of Computer and System Sciences"
MACRO scp "Science of Computer Programming"
MACRO sicomp "SIAM Journal on Computing"
MACRO tocs "ACM Transactions on Computer Systems"
MACRO tods "ACM Transactions on Database Systems"
MACRO tog "ACM Transactions on Graphics"
MACRO toms "ACM Transactions on Mathematical Software"
MACRO toois "ACM Transactions on Office Information Systems"

MACRO toplas "ACM Transactions on Programming Languages and Systems"
MACRO tcs "Theoretical Computer Science"
FUNCTION sortify purify"l"$change.case$
FUNCTION chop.word 's := 'len := s 1 len substring= $slen1 + global.max$ substring'sif
FUNCTION format.lab.names 's := s 1 "vv ll, jj, ff" format.name't := $tget.str.lang'name.lang$:= $name.langlang.en = t1"vv\ ll"format.name$ t 1 "llff" format.nameif s num.names1 > $bbl.space * citation.et.al*'skip$ if
FUNCTION author.key.label author empty$keyempty$ cite13$substring$ 'key if$authorformat.lab.namesif$

FUNCTION author.editor.key.label author empty$editorempty$ key empty$cite$ 1 3 substring'$keyif$ editor format.lab.names if$authorformat.lab.namesif$
FUNCTION author.key.organization.label author empty$keyempty$ organization empty$cite$ 1 3 substring"The"4$orgar
if'$keyif$ author format.lab.names if
FUNCTION editor.key.organization.label editor empty$keyempty$ organization empty$cite$ 1 3 substring"The"4$organi
if'$keyif$ editor format.lab.names if
FUNCTION calc.short.authors type"$book$" = $type$ "inbook" = or 'author.editor.key.label type"$collection$" = $type$ "proceedings" = or editor empty$not'editor.key.organization.label'author.key.organization.labelif$ 'author.key.label ifif 'short.list :=
FUNCTION calc.label calc.short.authors short.list "(" * format.year duplicate$empty$ short.list key field.or.null = or pop""'$skip$ if$*'label$:=
INTEGERS seq.num
FUNCTION init.seq 0 'seq.num :=
FUNCTION int.to.fix "000000000" swap$int.to.str$ * -1 10 substring
FUNCTION presort set.entry.lang set.entry.numbered show.url show.doi check.electronic calc.label label sortify " " * seq.num 1 + 'seq.num := seq.num int.to.fix 'sort.label := sort.label * 1 entry.max$substring$ 'sort.key:=
STRINGS longest.label last.label next.extra
INTEGERS longest.label.width last.extra.num number.label
FUNCTION initialize.longest.label "" 'longest.label := 0 int.to.chr'$last.label$:= ""'$next.extra$:= 0'$longest.label.width$:= 0'$last.extra.num$:= 0'$number.label$:=
FUNCTION forward.pass last.label label = last.extra.num 1 + 'last.extra.num := last.extra.num int.to.chr'$extra.label$:= "a"$chr.to.int$ 'last.extra.num := "" 'extra.label := label 'last.label := if$number.label$1+'$number.label$:=
FUNCTION reverse.pass next.extra "b" = "a" 'extra.label := 'skipif extra.label 'next.extra := extra.label duplicate$empty$ 'skip""$swap$ * "" * if'$extra.label$:= $labelextra.label *' label$:=
FUNCTION bib.sort.order sort.label 'sort.key:=
FUNCTION begin.bib preamble$empty$ 'skip$preamble$ write$newline$ if"$thebibliography$"$number.labelint.to.str$ * "" * write$newline$ "[1]1" write$newline$ "[1]1" write$newline$ "" write$newline$ show.doi "[2]2" write$newline$ "[1]https://doi.org/11" write$newline$ 'skipif
FUNCTION end.bib newline"