

Chess macros for chess games and puzzles

Marina Yu. Nikulina

St. Petersburg Technical University

St. Petersburg, Russia

marishka@ptslab.ioffe.rssi.ru

Alexander S. Berdnikov

Institute of Analytical Instrumentation

St. Petersburg, Russia

berd@ianin.spb.su

Abstract

The macro package `UCHESS.STY` described here generalizes the well known chess macros `CHESS.STY` by Piet Tutelaers [1], `BDFCHESS.STY` by Frank Hassel [2] and the less known `CHESS.STY` by Tomasz Przechlewski [9]. It adapts international (European) chess notation for other languages (primarily for Russian) as it is required by modern \LaTeX . It also allows the User to annotate non-classical chess games — hexagonal chesses, Ω -chess, etc. Special care is taken to allow the User to represent various chess puzzles with non-standard chess rules.

Introduction

The well known and available on CTAN chess macros `CHESS.STY` (by Piet Tutelaers) and `BDFCHESS.STY` (by Frank Hassel) are excellent but do not work well with the Russian language without manual correction of the original macros. Moreover, if the `BABEL` package (versions 3.6 and 3.7) is used, neither `CHESS.STY` nor `BDFCHESS.STY` can recognize the presence of `BABEL` and hence cannot support multi-language chess notation as they should. The other reason to issue a new macro set is that `CHESS.STY` and `BDFCHESS.STY` only deal with the classical chess game—such things as non-standard figures and non-standard chess boards are not supported.

The new macro package `UCHESS.STY` outlined here should correct these gaps. It follows the scheme suggested in `CHESS.STY` and `BDFCHESS.STY` and is more an extension of the ideas suggested in these packages than an original macro package. It includes support of multilingual chess notation in a more robust way (including support of the Russian notation). It supports multiple chess boards and positions simultaneously and contains special macros which simplify the presentation of chess problems. It also has special commands which help to represent non-standard chess puzzles with unusual boards and rules. It is based on an extended chess font which permits representation of chess and checkers on non-standard boards and with non-standard figures (like

the hexagonal chess [10] and Ω -chess [7] games), and will draw the arrows showing moves, etc.

Problems with foreign language support

One of the most powerful features of `CHESS.STY` by Piet Tutelaers is the support of chess notation for foreign languages. But since that part of \LaTeX changed greatly since `CHESS.STY` was released, the feature has steadily transformed itself to a weak point of the package. Let us consider this effect in more detail.

First, the support of foreign languages in `CHESS.STY` is entirely based on `BABEL`. But it recognizes the presence of `BABEL` by the existence of the macro `\babel@core@loaded`—which is obsolete since `BABEL` 3.6a (1996/11/02), as explained in the `BABEL` manual. As a result `CHESS.STY` loads `english.sty` even if `BABEL` is already loaded. (And if by chance it reads the obsolete version of `english.sty` included among other files with the bundle with `CHESS.STY`, the result may be rather strange.)

Second, the `CHESS.STY` support of foreign languages analyzes the current `BABEL` language each time you switch to chess mode and unconditionally redefines the chess notation as specified by this language. So it is impossible without extensive language switching to have chess notation in English while the main text is in German, French, Dutch or Russian. Moreover, if the corresponding language is

not supported by CHESS.STY at all (the current version 1.2 recognizes only `english`, `french`, `german` and `dutch` as valid languages), you'll see that your chess notation disappears without any warning or error message—so you must switch language to some known one before each chess fragment. (This defect is partially corrected in BDFCHESS.STY—namely, you can specify one language for chess notation and another language for the main text. But if the language selected for chess notation is not supported by CHESS.STY, you'll still find all your chess figures disappearing from your document without any warning.)

You can correct this defect by adding the support of the desired language in CHESS.STY directly. But if you would like to do this, you'll need to correct the original file or to redefine its internal macro `\select@pieces`—both operations are not too difficult for T_EX experts (especially taking into account the that the procedure is described in the documentation supplementary to CHESS.STY). But for the ordinary user it is not so clear how to do it, and there is no ready-to-use intrinsic mechanism for adding a new language in CHESS.STY with modest effort and in a standard way. (And the strange behaviour of CHESS.STY is quite annoying for an ordinary user because he/she cannot understand what happens and what is wrong with the document.)

Finally, CHESS.STY supports foreign language notation in the assumption that in *any* language one-letter notation is used. This is true for English, French, German, Dutch, Italian, etc., but in Russian it is not so: *two-letter* notation is used to distinguish *King* ('Kp') from *kNight* ('K'). Taking into account that for the Russian language there is no unified input encoding (different platforms uses different encodings for Russian letters) the problem how to support Russian chess notation in a unified way becomes even more difficult.

Foreign language support in UCHESS.STY

The support of foreign language notation is the main component which is different in UCHESS.STY and CHESS.STY (to be honest, correct support of Russian chess notation was the main reason to substitute CHESS.STY by UCHESS.STY). You may control the language used for chess notation independently of the language used for the main text by the command `\chesslanguage`. And although your chess language is synchronized with the main text if you specify the chess language as `babel`, synchronization is not obligatory.

The procedure for adding new languages into UCHESS.STY is modified as well. First, when a

special language is required for chess notation, the package checks that a characteristic macro command based on the language name already exists. If the command does not exist, a warning message is issued and the chess notation is switched to English (the default notation built-in for UCHESS.STY). Second, if you need to add a new language for UCHESS.STY, you just define the necessary macro command—this is enough to extend the set of languages supported by UCHESS.STY—you do not need to change anything thing inside the original UCHESS.STY commands. Finally, when you load the package UCHESS.STY by the L^AT_EX 2_ε command `\usepackage`, you can define the list of supported languages as its options—for each unknown option *name* the file *name.cld* is loaded (.cld stands for *chess language definition*). (UCHESS.STY also checks that the macro with the desired name is really defined in this file. For all main languages the corresponding .cld-files are included with the package UCHESS.STY.)

But the problem with the Russian language and its numerous encodings still remains. To solve it we added an additional (optional) parameter which helps the language switching macro to select the appropriate input encoding. Additionally, to simplify the user's work on defining the language switching macro there are standard commands which help to define one-letter and two-letter abbreviations for chess figures. And if by chance the language switching mechanism misbehaves (as it does for CHESS.STY where the chess figures just disappear from your text if there are no standard chess abbreviations for your language), the user can always use macro commands with fixed names.

The UCHESS.STY support of other languages also allows the user to switch flexibly between alphanumeric notation and the 'figurine' notation in the document without major changes to the input file (where the alphanumeric notation is in agreement with your language). Special precautions are performed to adapt other special symbols used by chess notation properly when the language is switched as well (for example, the symbols used for international *Chess Informant* notation and the notation traditionally used in Russia, Poland, etc., differ widely as concerns the symbols used for check, checkmate, stalemate, etc.).

Non-standard chess games

Chess board for 4 players and its rules It is nice that there is the support of the classical chess game in L^AT_EX, but today the non-standard chess games are becoming more and more popular [8, 7].

While it is necessary to keep chess figures of *four* colors for the chess game for 4 players [8], it is possible to overcome the restriction of typical black-and-white typography by rotating the corresponding chess figures by 90° , 180° and 270° . The fonts described in [3] enable us to generate the rotated fonts without any problems but it is questionable whether that it is necessary to support such an exotic chess game inside UCHESS.STY. Maybe it will be done someday — if somebody really needs it.

Ω -chess board and its rules While the chess game for 4 players [8] seems to be just a private joke for home entertainment, Ω -chess [7] is a more serious game, and its support by L^AT_EX is a desirable feature. Support of the Ω -chess board requires two additional figures (*wizard* and *champion*), a non-standard chess-board with 104 fields (10×10 plus four separate corner squares) and support of non-usual moves and field notation in the way that is already done for the classical chess game in CHESS.STY. Additional figures are included in our new chess font [3], and the extended chess notation is supported when you mark your chess board as an Ω -game by a special option.

Hexagonal chess boards and their rules In addition to the Ω -chess board there are (at least) two hexagonal chess boards [10] with chess rules of their own. These boards are composed from hexagonal fields colored in three colors (white, gray and black), and although *standard* chess figures are used, the initial positions and the rules how to move the figures over the board are far from being standard. As for Ω -chess the support of hexagonal chess games is included in UCHESS.STY although serious modifications of the internal representations used for chess boards in CHESS.STY were necessary.

Chess problems The classical chess game is divided into two relatively independent branches. There is the chess game itself where two people fight for victory, and there is the solution of specially prepared chess problems. (Actually there is a third branch as well — the computer chess game and computer chess analysis — but these are not too different from the other two branches as concerns chess typesetting.)

While the support of an ordinary chess game is done in CHESS.STY in a user-friendly manner, and while BDFCHESS.STY supports playing chess games by post in the same way, the support of chess typesetting for chess problems is not so flexible. To assist the user working in this specific field UCHESS.STY contains the support of several boards and positions simultaneously, storing and copying the board

content, transformation of boards, adding/removing individual chess pieces, etc. There are also special chess environments used to typeset the main stream of the solution, its side streams and the alternative variants — see section ‘Additional environments’ for more details.

Puzzles, fairy chess games, checkers, etc.

Apart from classical chess games and chess problems there are so-called unusual (fairy) chess games and problems — i.e., with non-standard rules, boards, figures, etc. While the typesetting of chess diagrams with non-standard rules is more or less covered by macros designed for ordinary chess diagrams, this is not the case when we deal with non-standard (for example, non-rectangular) chess boards and unusual chess figures.

Macros from UCHESS.STY extend the flexibility of defining non-standard rectangular and hexagonal chess boards. In CHESS.STY each field may be black or white, occupied by a chess figure or not, but in UCHESS.STY two more variants are included:

- empty field which is outside the board (drawn as an empty square),
- empty field which is a hole inside the board (drawn as a square filled by solid black color).

While neither field can be occupied by a figure, they are drawn in a different style for a clear reason: it is not reasonable to fill with black the entire region outside the non-rectangular board, and it is difficult to distinguish the hole inside non-rectangular board from the white field if it is just empty without an additional marker.

The other aspect essential for fairy chesses is the existence of non-standard figures. While there are no strict rules how to represent these figures graphically (the Ω -chess game is an exception [7]), it is more or less clear that they should be different from the ordinary chess figures. To satisfy the needs of this group of users, UCHESS.STY supports additional pictograms (crosses, bullets) to represent unusual chess figures, and a big bold circle which can modify the meaning of an ordinary chess figure if it is necessary. And in addition to these chess-like figures it is necessary to note, that *checkers* and *checker-like* board games can be considered as fairy chess games as well — so it is natural to include support of checkers in UCHESS.STY also.

Finally, it is necessary sometimes to draw markers, borders, arrows, etc., over chess boards (particularly for child-level chess textbooks). A separate font enables us to draw arrows, etc., over the chess board in the same way as in the `picture` environment, and it is possible to add plain borders and to make

bold boundaries between the square and hexagonal cells of the board.

Extended chess encoding

As a result of our attempt to cover all these contradictory requirements, a special chess font (or, more strictly, chess encoding) was developed. It includes all classical chess figures (black and white) and some special-purpose figures (wizard, champion, checker, double-checker, crosses and bullets) are also added. Non-symmetrical figures exist in two variants, straight and mirrored (hopefully there are just two figures of that kind, *knight* and *wizard*)—because, for example, it is recommended to use the horse with the head rotated to the left when it represents the knight on chess board, and to the right when it represents the knight in chess text [2].

Editor’s note: Surely the figures don’t exist as a text character, but rather as some sort of character code... We’ve already been told that in Russian notation there is a two-letter representations.

Each figure exists as a text character, as the figure placed over a black or over a white square field, and as the figure placed over a white, black or gray hexagonal field. (In fact, there is *no* separate figure to be placed over a white hexagonal field—it is enough to have the figure placed over a white square and a simple T_EX macro. But the difference between ‘text figures’ and figures placed over white square fields is essential—see [3] for more details.) In addition there are the empty square and hexagonal fields of corresponding color, the fields filled by a solid black color, the lines used to surround the square field or the hexagonal field by a frame (or to make the solid frame around the whole board), and the solid circle used to modify the meaning of a chess figure.

The final chess encoding and the details of new chess fonts are described in [3].

Chess Informant notation

Aside from chess figures there are many special symbols used to annotate real chess games in an abbreviated manner. The most famous is the international *Chess Informant* system, but there are also other symbols locally accepted in national chess publications. Since these symbols can be used in text only, there is no reason to put them in the same font as ordinary chess figures. Moreover, these symbols should be visually compatible with the ordinary Computer Modern text and, hence, change their style as the style of the main text is changed. L^AT_EX 2_ε has a sophisticated but user-friendly NFSS-system which controls such behaviour

nearly automatically. All what we need to do is to fix the list of necessary symbols and the font encoding, to create the fonts for all necessary variants (size/shape/series/family) and, finally, to prepare corresponding .def- and .fd-files and define macros.

Since this part of UCH_{ES}S.STY is under preparation now, it does not contain symbol fonts which are compatible with Computer Modern and European Modern and the representation of the notation of *Chess Informant* to a full extent, but there are definite plans to extend the support of the international chess notation to the level where *Chess Informant* notation and national notations are fully supported as well. (It is worth noting that sometimes *Chess Informant* notation and national notations are different—as a result the language-switching commands should control this difference as well.)

Arrows, borders, etc.

Chess figures and chess symbols described in sections ‘Extended chess encoding’ and ‘*Chess Informant* notation’ are enough for almost all professional chess typesetting. But what about textbooks? Quite often it is necessary to draw the arrows showing the moves of particular figures, to emphasize some fields by special markers, to draw a special border around a chess diagram, to mark explicitly the boundary between two chess fields, etc.—and there are no such elements in our fonts nor macro commands at our disposal!

But this just means that such elements and such macros should be added. Hopefully, there is a good predecessor for this—namely, the L^AT_EX `picture` environment based on restricted LR-mode which enables construction of arbitrary two-dimensional mosaics from discrete elements by explicit specification of individual two-dimensional coordinates. Our environment allows the user to annotate by arrows, markers, borders, bounding lines, etc., for textbook chess diagrams that are organized in the same manner.

Although just now there is no special font with arrows, etc., available to UCH_{ES}S.STY for that purpose (and, consequently, there are no macros to draw arrows over the chess board), we have plans to add the necessary extensions.

Multiple chess boards

An essential extension of CH_{ES}S.STY is the support of multiple chess boards inside the same document. The board is identified by its name, and by default the board `current` is used. The board contains the description of the current position, the flags showing the state of the game (white or black side is to make

the move, is it possible to castle, etc.) and a special flag identifying the rules of the game. The board can be of non-standard size (square or rectangular) and, besides ordinary fields, can contain special fields—namely, *empty* fields which are outside the board and are displayed as empty white squares (hexagons), and *null* fields which are displayed as black squares (hexagons) and represent the holes inside the standard board. It is possible to make a copy of the current content of a board with a new name, to correct the content of the board in a silent mode without actually typesetting the moves, to change a board’s state manually, etc. The boards obey standard TEX rules about block nesting which enables the user to play the game variants and to investigate side game branches without any problems and special precautions.

Additional environments

Is it true that now we have *everything* we need for accurate chess typesetting? Not at all—we need specialized *environments* to mark up chess-oriented text fragments and to mark up logically different chess-oriented material as well. A very good example can be found in the documentation of `CHESSTY`—namely, the case where the main stream of the annotated chess game is typed in bold while the alternative variants of the chess game are typed as ordinary text. While these environments for logical mark-up are outside the main style created by Piet Tutelaers, they are an intrinsic component of our package `UCHESSTY`.

Acknowledgements

We express our warmest thanks to our colleagues from other countries—their careful attention supports our activity in TEX even in such unfavourable conditions as they are in modern Russia.

Special thanks are to Bogusław Jackowski and Tomasz Przechlewski for valuable consultations, lit-

erature references and ready-to-use chess fonts and chess macros used when we worked on `UCHESSTY`.

Alexander Berdnikov would like to thank separately Dr. A. Compagner from Delft University of Technology for his long-term friendship and patient attention as a teacher—not necessarily relating to some TEX or $\text{L}\text{A}\text{T}\text{E}\text{X}$ joint works.

References

- [1] Piet Tutelaers. A font and style for typesetting chess using $\text{L}\text{A}\text{T}\text{E}\text{X}$ or TEX . Machine-readable document (supplement to `CHESSTY`), 1992.
- [2] Frank Hassel. A $\text{L}\text{A}\text{T}\text{E}\text{X}$ style for management of correspondence chess games. Machine-readable document (supplement to `BDFCHESSTY`), 1993.
- [3] Marina Nikulina, Alexander Berdnikov. Chess fonts for chess games and puzzles.—in these Proceedings.
Editor’s note: This paper was not in fact presented, and will not appear in the proceedings. Please supply an alternative description.
- [4] Zalman Rubinstein. Chess printing via `META-FONT` and TEX . *TUGboat*, **10**, pp. 170–172 (July 1989).
- [5] David Tofsted. An improved chess font. *TUGboat*, **11**, pp. 542–544 (November 1990).
- [6] Jan Eric Larson. A chess font for TEX . *TUGboat*, **10**, pp. 351–352 (November 1989).
- [7] Ω -Chess Home page: www.omegachess.com
- [8] Chess game for 4 players: www.4playerchess.com
- [9] Tomasz Przechlewski, `CHESSTY`, private communication.
- [10] Wojciech Pijanowski. Gry w które grałem. “Pomorze”, Bydgoszcz, 1989.
- [11] Е. Я. Гик. Занимательные математические игры. “Знание”, Москва, 1987.