Acta Ethnographica Hungarica, 60 (1), pp. 121–170 (2015)
DOI: 10.1556/022.2015.60.1.14

COMPUTER-AIDED DANCE ANALYSIS IN PRACTICE – WITH LABANATORY

Gábor Misi

Jászhalom u. 27, H-1163 Budapest, Hungary
E-mail: misigabor@yahoo.com

Abstract: Hungarian folk dance has strict structure, which plays an important role in its authenticity. Besides the connections of dance elements, it is important to consider quantitative features of a dance such as the frequency of dance elements and the frequency of connections. The number of occurrences of a dance element or several consecutive dance elements in a dance performance can be counted with a computer tool if the dance is notated and its notation is digitized.

Labanatory has been developed in Hungary as a tool that is capable of searching for repeating patterns in Labanotation scores, that is to identify recurring movements. The paper shows the software in use: how the researcher can use this kind of tool in practice. It shows actual examples of detecting recurring notated movements with several kinds of searches (simple, compound, wildcard, symmetric, and augmented searches). It points out that searches have to be used in an iterative way during the analysis when creating 'good' queries from Labanotation symbols. Notation parts can be labeled in the program in any phase of the analysis and the researcher can revise the label markers offered as a result of a search. From the labeled dance parts, the user can make basic statistics on the analyzed dance performances.

The examination of original dance texts is very important in dance education. It is beneficial if a student can access fieldwork-texts directly, not only through the teacher's filter. The student can focus on dance elements in accordance with his own interest by creating his own queries and performing the related computer searches. He can detect dance elements of any length and examine any consecutive combinations of them. Evaluating search results can allow the student to interpret dance structures in new ways.

Keywords: dance notation, Kinetography Laban, Labanotation, dance analysis, retrieval, software, query, search, label, statistics

INTRODUCTION

Dance analysis has always been inspired by linguistics. To remain objective, however, only those linguistic methods can be adopted in dance research that ignore the meaning of language elements, since dance elements do not have that kind of meaning. These methods are the formal methods.

Generative linguistics must handle language elements in a purely formal way.[1] From this viewpoint, a word in a sentence must be considered only in terms of its position and not its meaning. Although the purpose of generative linguistics is to describe language competence generatively, it has to examine the positions of various elements in the performance that is in a corpus of the language.

Corpus linguistics, which is a growing discipline today, studies the spoken or written text corpus alone.[2] Its approach is that original texts should serve as examples of grammatical rules. Corpus linguistics mainly uses computer programs with text search function for research.

Since search function is also available in dance applications[3] for notated dance[4], dance research can use these programs for formal dance analysis.

This paper discusses Labanatory 1.0 software. After an overall review of its features, a few exercises will illustrate how the program operates in practice and how a researcher or student can use the software to examine dance texts. Obviously, the precondition of a computer-aided dance analysis is that the dance in question has already been notated.

USER INTERFACE

Labanatory is an application for AutoCAD R14 vector graphics software. Besides the standard commands, Labanatory defines special commands on Labanotation for the user. After loading the application, the new commands can be executed from the text window in the lower part of the screen, just as the standard AutoCAD commands. The elements of AutoCAD user interface, hotkeys, menus and toolbars are also built on the new commands as well.

Although the user can access a function in several ways (e.g. a Labanotation sign can be moved to the left with the `'move _ sign'` command, Ctrl+← hotkey, Edit/Move/Left menu item, or with the first icon of Edit toolbar), only the most convenient will be described for each function below, always in parenthesis. The full description of all the commands is given in the Appendix. The menu layouts are also shown in the Appendix *(Figures 1–8).*

The number of dots in a menu item name informs about the input and output of the related command. If it contains three dots, a dialog box will appear where the user can select the parameters for the execution. In the case of two dots, the user has to type the parameters in the text window. If there is one dot or no dot at all, no parameters are expected. One dot indicates that the output is not displayed in any graphic window but in the text window.

The functions will be presented in a logical order on the basis of their use. After a short review of data input functions, search functions will be described in more detail. At the end, practical exercises will be given.

[1] Chomsky 1957.
[2] Hunston 2002.
[3] Fügedi 1995; Misi 2002.
[4] Szentpál 1976; Hutchinson 1977; Knust 1979.

DATA INPUT

After loading the application (File/Load Labanatory), you have to open a file where the scores are stored. The file can be new (File/New) or previously created (File/Open...) if you want to continue your work. Although only one file can be open in AutoCAD R14, a file can contain several staves.

When you create a new staff (Staff/Add/Normal..), you have to give a unique name to it, and certain size values (length of starting position, length of upbeat, number of beats per measure, number of measures, distance of numbering captions). The command draws the lines of the new staff and zooms to the staff; you can see its name and the measure numbering on the left (see Appendix *Figure 12*).

The unique name of the staff has to be used for further commands to identify the staff. You can list all the staves stored in the file (Staff/List.). You can rename a staff (Staff/Rename..), modify its size (Staff/Modify..), delete it (Staff/Delete/a Staff..), or delete only the signs belonging to it (Staff/Clean..). Deleting several staves at the same time is possible by typing a name mask with asterisks (Staff/Delete/in Cycle..).

Working on staves can be made easy with the following functions. You can zoom in on a staff in a window (Staff/Zoom/to Staff..), or browse among the staves (Staff/Zoom/in Cycle..). You can split the graphic window into several parts (View/Split/, 'vports'), hide or show various parts of staves: support line, gesture lines, measure lines, tick marks, the staff name, measure numbering, beat numbering, bookmarks and markers (View/Staff parts..).

Editing the drawing can be more convenient if the column-grid is displayed (F7), or the sign placement is fit to a column (Edit/Set placing snap/to Column), half-column (F12), or nothing (F11).

You can place Labanotation signs in the drawing with the aid of toolbars (Appendix *Figures 9* and *11*). First you have to select the required sign icon, click it, then click with the mouse pointer on the required place in the drawing. Then the sign appears in a default size. Next you can move the placed sign in any of the four directions (Ctrl+arrow keys), and you can set the step distance of moving (Edit/Set moving step/). You can resize the sign into properly restricted directions (Ctrl+arrow keys of the numeric keypad or the wide arrow icons of the Edit toolbar). You can transform the sign if it is a special one: in the case of path or directions signs, gaps can be created in the middle of the sign (Ctrl+PageUp, Ctrl+PageDown). Finally, you can delete the sign (Ctrl+Shift+D).

A little background information must be explained here about the drawing mechanism in Labanatory. Labanotation signs consist of several lines. There are auxiliary lines (e.g. 45 degree lines of a rotation sign, 45 degree lines inside a high level direction sign, vertical lines inside a path sign, etc.), which are erased and redrawn in accordance with the new position or new size after a modification. If you move a sign across another sign, some auxiliary lines can be erased but not redrawn. In that case you must redraw certain signs (Ctrl+Shift+A) or the full drawing (View/Refresh).

It is useful to move, delete or redraw more signs at the same time. For this purpose you can create a set of selected signs (selection with mouse pointer, then Ctrl+Shift+S). Each element of this sign set will be the subject to the next sign operations. The sign se-

lection set is overridden when you place a sign: from that point the sign selection set will only contain the last placed sign. It was designed this way because sign operations most frequently relate to the last placed sign.

The sign selection set is useful for performing copy-cut-paste (Ctrl+Shift+C, Ctrl+Shift+X, Ctrl+Shift+V); you can even perform these operations through a given file (Edit/Copy to File..., Edit/Cut to File..., Edit/Paste from File...).

When you finish editing, do not forget to save your work (File/Save, File/Save as...) before you exit from the program (File/Exit from AutoCAD).

SEARCH AND SEARCH OPTIONS

The next group of commands is related to searches and these commands operate similarly to search facilities of text editors. You can construct the sign pattern that you are looking for in a given staff. First you have to create a special 'query' staff (Staff/Add/Query..), and you have to insert the required sign or signs into the relevant columns of this staff.

In the background during the search process, sign pattern matches are examined in a matrix representation,[5] which is created on the basis of cells in a certain grid. You can display the grid on the screen (F7, Appendix *Figure 13*), and if you are an advanced user, you can change the vertical size of the grid (`'grid'`) before the execution of searches.

When you start a search (Search/Simple search..), you have to identify first a query staff, after this the staff in which you want to find the query pattern, and in the end, you have to accept or change the default values of the search parameters.

As a result of a search, the found signs are displayed in color. You can change the color either before or after the search execution (Search/Set hit color/).

You can browse among the hits: you can go to the first, the previous, the next or the last one (Search toolbar, four icons in the middle, *Figure 14*) as long as the hits are colored, that is the search result has not been cleaned. You can clean it manually (Search/Clean search), and it is cleaned automatically when you start a new search.

The command line search parameters determine what kinds of hits are produced. The first parameter determines what length you want to search the pattern: only original, only augmented *(Figure 15)* or both lengths. The second parameter determines what instances you want to find: only identical, only symmetrical *(Figure 16)* or both. The third parameter determines the timing of the hits relative to the measure lines (on measure, half-measure, between them, or any timing, *Figures 17, 18*). The fourth parameter determines whether the lengths of strengthenable signs are considered in the examination of pattern matches or not, that is the match is sign-length tolerant or not.[6] If you accept the default values of all the four parameters, the hits will have original length, identical symmetry, any timing, and sign-length tolerant match. The match is always pre-sign tolerant because of the matrix representation applied by the search algorithm.[7]

[5] Misi 2008.
[6] Misi 2012.
[7] Misi 2008.

You can also execute compound searches (Search/Compound search..) by constructing a compound query where sign patterns of several query staves are connected with logical operators (OR, NOT). In this case you execute a search with the query pattern that corresponds to the evaluated logical expression of the given query patterns.[8] There is no need to use AND operator, since creating only one query pattern with several signs for a simple search means that you want to find the signs together. In the case of alternative search (OR search), those patterns will be produced as hits where any of the given query patterns matches *(Figure 19).* In the case of exclusive search (NOT search), you have to give at least one exclusive query staff, and only those patterns will be produced as hits where the exclusive pattern does not match, giving a filtered search result.

You can avoid using OR operator if the difference between the alternative patterns stands on just sign alternations in certain Labanotation sign groups. The software includes a few predefined wildcard signs (colored blue, Wildcard toolbar, *Figure 10*); each of them represents a Labanotation sign group. If you place a wildcard sign into your query staff, the search will produce all the patterns as hits where the query pattern matches after replacing its wildcard sign with any of the signs from the related sign group. You can also use wildcard signs in a compound search, in any part of the logical expression (*Figure 21* shows an exclusive search with a wildcard).

SPECIAL FUNCTIONS

You can add markers to the colored hits one by one or together (Search toolbar, last row, first and second icons). A marker consists of a line and a text, and it appears beside the staff on the right *(Figure 22).* The marker text has a default value, based on the last executed query, but you can set any text as current (Search/Set hit marker text..). You can set the current marker line type and the current marker distance (Search/Set marker properties..). Additionally, you can put markers beside the staff at any location, with any length, with any text, regardless of the search results (Search/Mark individual..).

You can delete the markers assigned to the colored hits one by one or together (Search toolbar, last two icons). You can also delete the markers with a given marker text that belong to a given staff (Search/Delete markers.), or all the markers that belong to all the staves (Search/Clear markers).

You can create statistics on the markers. You can list the marker texts and the frequency with which the markers can be found in a given staff (Tools/List marker frequencies.., *Figure 23*). You can also list the locations where the markers with a given marker text can be found in a given staff (Tools/List marked parts.., *Figure 24*).

Two special commands have been developed specifically for dance analysis, focusing on marked staff parts that correspond to the motivic dance elements in the analysis.

1. Making a motif catalogue: if you have several staff parts marked with the same marker text, you can copy the corresponding sign sets into new 'cut-off' staves (Tools/Collect marked parts.., *Figure 25*). You can determine whether the sign sets for the copy

[8] Misi 2012.

operation should include signs inside certain grids under the marker, considering movements with upbeats.

2. Discovering the nucleus of a motif: if you have some 'cut-off' staves containing certain sign sets, you can obtain the common signs from them in two ways in particular: by creating the intersection or sign-majority intersection of Labanotation signs (Tools/ Project common parts, *Figure 26*).[9] When you list the staff names in a drawing (Staff/ List.), you can also filter for 'cut-off' or 'intersected' staves.

The Labanatory functions have been reviewed in a nutshell. The next chapter describes the use of the commands in practice.

PRACTICE

The following dance analysis example will include the tasks of a complete computer-aided analysis. The dance example is a Transylvanian male solo dance.[10] Three periods of the dance have been simplified to create the learning corpus (Appendix *Figure 27*). The Labanotation score uses unit-timing indication.[11]

The purpose of the analysis is to find and mark the recurring dance elements,[12] where the markers cover as large part of the dance as possible. The proper queries that ensure the optimal coverage will be constructed gradually. You can follow the various steps of the analysis with the aid of *Figures 27–48* in the Appendix. The work exercises have been designed to be read as follows: first read the question (Q), think about it, and check the related figure only when you are reading the answer (A). The marker texts in the figures will probably seem strange because they are quite short. These dance element names consist of one letter per beat, referring the movement type of the relevant beat. What matters here is that the marker texts are unique query by query.

Q1. Look at *Figure 27,* especially the recurring parts at the beginning of measures 2–6. What query should be constructed to find these parts? Use a one beat long query staff for the time being.

A1. Place just a single heel sign on an imaginary touching leg gesture into your query staff. Execute the search to find both identical and symmetrical instances on measure start. This parameterization will also be used for the next searches. You can see the marked hits in *Figure 28*. Observe that there are also hits in measures 10, 12 and 18.

Q2. Examine measures 3–4, where no hits have been produced. You should decide whether touching the floor with full heel or half heel is the same movement in your analysis. If yes, how can you find both touching leg gestures in only one search? If you find them in the same search, you can mark them at the same time.

A2. Construct a compound query from two query staves. One of them has to contain

[9] For definitions see Misi 2005, algorithm descriptions Misi 2005 and Misi 2008, examples Misi 2005: Appendix 7.

[10] Misi 2005: Appendix 3.

[11] Fügedi 2008.

[12] Misi 2012.

a full-heel sign, the other a half-heel sign. Connect them in an OR search. You can find both kinds of touching leg gestures using the OR expression. *Figure 29* shows that nine hits are produced.

Q3. How can you construct a simple query to produce the previous hits?

A3. Use a wildcard sign for any part of the sole. This way it is sufficient to use only one query staff for a simple search. *Figure 30* shows the search result. Observe that a sole touch is also produced as a hit (measure 8), in accordance with the meaning of the wildcard.

Q4. The previous searches only examined whether the touch was performed. The touching leg gestures were not distinguished in terms of direction, leg bending or dynamics. If you were to decide that any of these features is important for your analysis, you should construct queries using direction, space measurement or dynamics signs. However, the focus of the analysis will now shift to other exercises to search more than one beat long dance elements. Study how the previous hits are continued. What queries can be constructed to obtain longer dance elements?

A4. You can see that some continuations are second positions (measures 4 and 5), while others are touching leg gestures with eights toe in a side direction (measures 3 and 6). At this point, two queries can be constructed: after the first beat, which contains the sole-part wildcard sign, write two side direction signs in the support columns of the first query staff, and write a side direction sign in the gesture column of the second query staff. You can execute searches with either the first staff, see *Figure 31,* or the second staff, see *Figure 32*. To find both movements, you have to execute a compound OR search instead of the simple ones, see *Figure 33*. Nine hits are produced, where seven instances have position continuation and two instances have touching leg gesture continuation. You have to make a decision in the analysis again, on whether the two kinds of side movements of the leg mean the same movement or not. In this particular analysis, the answer is 'yes', and therefore the next queries will be compound OR queries.

Q5. Can you simplify the previous compound OR query by using wildcard signs?

A5. No, it is not possible, because the alternative signs are placed in different columns of the query staves.

Q6. Can you find more than two beats long dance elements at the places where the previous hits were produced? What signs can you use in the third beat of the query staves?

A6. The two beats long dance elements are continued with a first position in almost every case. Verify this with a search. Construct a query using two place direction signs of the first position indication in the third beat of the query staff, or more precisely, both query staves of the compound OR search. After the search execution, all nine previous hits are produced again, see *Figure 34*. No decision is needed at this time, and in fact the search result suggests that it is worth taking three beats long dance elements rather than two beats long ones, because they cover more beats of the whole dance score.

Q7. What about four beats long dance elements? Can you identify these by lengthening the previous queries? What queries can you construct to obtain them?

A7. If you examine the next beats, you can see that the next movement is a step (measures 2 and 3), or a jump to position (measures 6 and 10). In this particular analysis

the decision is that these movements are different and they should not be marked with the same marker. Therefore, there is no need to execute a compound OR search now, and it is sufficient to construct two query staves for two simple searches.

Q8. How can you find one-leg support and two-leg support in the fourth beat separately in your searches?

A8. It is easier to construct a query for searching a two-leg support, which is actually always a second position. Use two direction signs in the fourth beat of the query staff in the same way like they were used in the creation of the second beat before. *Figure 35* shows that five hits are produced. Constructing a query for searching one-leg support is more complicated. The one-leg support in the fourth beat is indicated with a side direction sign sometimes (measures 2 and 5), and with a place direction sign at other times (measures 3 and 4). If you decide that these movements will be not distinguished on the basis of their directions, use a wildcard direction sign. Write the wildcard direction sign into both OR query component staves. If you execute this search, it will unfortunately also find the two-leg supports, because the wildcard direction sign of the query pattern will match one of the two direction signs of the position indication. The query is only correct if a NOT component is added. Create a third staff, and write just a direction sign on the other leg in the fourth beat, see *Figure 36*. The search in accordance with this compound expression finds all patterns where the support is one leg in the fourth beat, and correctly, it does not find the positions.

While the previous search result suggested that it is worth taking three beats long dance elements, because they ensured a greater coverage of the whole dance, this result does not suggest that identifying four beats long dance elements is the best solution. The same nine measures coverage is achieved with two kinds of elements, and the distribution of these elements is near to an even distribution due to the four and five hits.

Q9. Now take a look at the movements from measure 9 that have not been examined yet. What queries can you construct so that the hits should cover these measures?

A9. The hand claps on the leg in the second beat of many measures. The query can be constructed with a retention sign in the first beat and a contact bow in the second beat, see *Figure 37*. The query staff does not contain any dynamics sign, pin or joint sign; this minimal query pattern is enough to produce hits exactly in the required measures. No distinction is made in terms of whether the support is one leg (measure 9) or a position (measure 11) in first beat, whether the next movement is a jump (measure 9) or not (measure 15), whether the weight moves to the side (measure 11) or remains in place (measure 9) in the second beat, or whether the clap is performed with two hands (measure 17) or one (measure 9). These differences can be identified using proper Labanotation signs in the query staff. It is up to the analyst to decide when two dance elements are interpreted as identical or different. The search with the present query produces hits in seven measures.

Q10. Examine how these claps are continued. Search three beats long dance elements. What query can you construct?

A10. The next movement is a step always. It is sometimes in a forward and sometimes in a place direction. Without distinguishing the step directions, use the wildcard direction sign in the third beat of the query staff. According to *Figure 38,* hits are produced in all the seven measures which were covered in the previous search. This result suggests that

constructing longer queries is useful here, because this allows you to cover three beats instead of two beats.

Q11. Can you find four beats long dance elements? Search for them on the basis of the possible continuations.

A11. The continuation of the clap and step is a new step (measures 9 and 17) or a new clap with the other hand (measures 15 and 21). You can construct two query staves using some of the signs that have been already used. The first query is presented in *Figure 39.* The search produces four hits, though one of them contains a step to position instead of one-leg support in the last beat (measure 11). The second query is presented in *Figure 40.* The search produces two hits, and both contain a clap on the thigh in the last beat. While the three beats long query produced seven hits, the four beats long queries produce only six hits, and there is a unique dance movement (measure 19) out of the seven previously covered measures. In spite of this, it may be worth examining further continuations of the two alternative dance elements.

Q12. What movement can you see in the fifth beat after the thigh claps in the fourth beats?

A12. The next movement is a clap again, where the hand claps on the opposite leg. *Figure 41* shows the query which you can find in both instances. The contact bow is drawn in the query staff carefully: its two ends are placed in the opposite hand and leg columns.

Q13. The fifth beat overhangs the measure length, and the cross-claps appeared in the first beats of the next measures. Consider if it is worth searching with a very short, one beat long query staff, using measure start parameterization to find these one beat long dance elements.

A13. If you execute the search in accordance with *Figure 42,* you can find one more instance of the cross-clap (measure 24). The benefit of searching in this way is that you can cover three measures instead of two. Examine the query staff. It is an already refined version due to a joint sign added. The minimized query pattern with only a contact bow is not precise enough, because the search on it would find a non-required clap: a thigh clap which is a part of a two-hand clap starting on measure (measure 20). If you look at an additional beat, you can discover further continuations: one out of three is a thigh clap (measure 22), while the other two have no movements in the next beat, therefore the rhythm is augmented (measures 16 and 24).

Q14. Now return to the parts where the clap in the second beat is continued with two steps, see *Figure 39.* What further continuations can you see there? What query can you construct?

A14. The two steps are continued with a movement sequence that was discovered at the beginning of the analysis. It was a one-, two- or three beats long dance element starting with a touching leg gesture. Concatenate the clap query pattern and the touch query pattern. *Figure 43* shows that the four beats long pattern is extended with a three beats long pattern in the query staff. The figure also shows the found seven beats long dance elements.

Q15. Examine why the last search did not produce a hit in measures 11 and 12, where the first measure contains a clap and the second measure a touching leg gesture pattern.

A15. While the clap is performed with the left hand, the touching leg gesture is performed with the right leg. The search did not find them together since both touching movements are drawn for the right limb in the query staff.

Q16. The last search result shows that it may be worth distinguishing hits on the basis of the side of the body on which a movement occurs. How can you put different markers to identical and symmetrical dance elements?

A16. Use the search parameters in the software. Execute two searches on the same query staff with different parameters to produce only identical instances in the first, and only symmetrical ones in the second. Set different marker texts before the two markings. *Figure 44* shows the search result after the two searches and markings. *Figure 45* shows similar double marking on another dance element. The previously used marker texts are extended with the postfix `_ j` or `_ b` depending on the body side, `_ j` for right and `_ b` for left. *Figure 46* shows all the instances of both dance elements and both sides. You can study the connections of the left and right instances in the figure.

The analysis could have been performed by distinguishing the body side from the beginning. The analysis could have been performed by using the same queries in a different order. The analysis could have been performed by identifying other dance elements, for example elements that do not coincide on measure start. The process of how the query patterns in this analysis were constructed is presented in *Figure 47.* The query patterns are progressively more precise from left to right, and they are increasing in length from the bottom.

The optimal query patterns were constructed with intuition in consideration of the layout of signs, that is which signs occur together or alone. The evaluation of the previous search results and the feedback was important. The queries were refined gradually with several tries in consideration of the distribution of the hits also. The general method of computer-aided analysis is that the user constructs several queries, executes searches with them, and evaluates the hits produced by them. If a query is not refined enough, and therefore the search result based on the formal pattern match is not good enough for the user to mark all the hits, the user can browse among the hits, and can mark them one by one, deciding which should be interpreted as a dance element. It is not necessary to mark all the hits after a search, though in the example analysis all the queries were constructed and refined in a manner so that all the hits would be marked, and the same search-and-marking process can be executed easily again later.

The purpose in an analysis under the above method is to cover as many parts of the notation as possible. The coverage cannot be complete in general, because usually there are certain dance parts that are not repeated. It is not worth constructing a query and executing the related search to produce an only hit. These unique parts can be marked one by one individually. However, it is not necessary to insist on covering every beat of the entire dance, and a few notation parts can be left unmarked. On the other hand, it is not necessary to attach only one marker to any beat of the notation; a beat can belong to various markers that overlap each other. Besides the search result based markers, *Figure 48* shows five individual markers for measures that have not been covered with searches, and a sixth individual marker for a two measures long part that already had two one measure long markers. This double marking shows quite well how a longer dance element can be

created from two shorter ones, and reveals how they can be connected. The frequency of the dance elements and the frequency of the connections of the dance elements can be examined with appropriate search executions and marker statistics. All these analytical examinations are supported by the Labanatory software.

AutoCAD supports command execution in batches. An advanced user can store a sequence of Labanatory commands that belong to a certain analysis in an executable script file as a scenario of the analysis. This way it is possible to re-run the whole analysis from the beginning. *Figure 48* is a result of a run script, and that script is also shown at the end of the Appendix.

The paper presented a brief overview of computer aided dance analysis by analyzing a short example corpus. The searches on dance notation were performed with a software application that has the capability to examine the matching of Labanotation sign patterns.

SUMMARY

Having a software like Labanatory, you can discover the recurring parts of a notated dance. As it was presented in the example analysis, it is worth executing several search tries in iterative steps to find the longer dance elements. Intuition has an important role in the method of constructing and refining queries. (Labanatory does not have a built-in algorithm that discovers the frequent patterns and suggests queries automatically.)

Computer aided dance analysis allows you to find the instances of a certain dance element (the equivalent of *lemma* in corpus linguistics), examine its each variant (*type* in corpus linguistics), and the environment of each variant. You can study compound motifs (*idioms* and *collocations* in corpus linguistics) by constructing new queries with concatenations of the related queries.

A non-repeating dance part, where only one hit would be produced by any search, is known as an invariant in ethnochoreology[13] and as a *hapax legomena*, or *hapax* for short, in corpus linguistics.

Using the terms lemma, type, hapax, idiom and collocation, dance research and corpus linguistics could use common concepts. On the other hand, dance research cannot adopt every method used in linguistics. The two disciplines have different problems to solve. There is no written word-separator in dance such as the space character used in corpus linguistics.

The approach of corpus linguistics and the use of software searches can be very effective in language or dance teaching. Monika Szirmai argues that corpus linguistics in language teaching would allow students to research corpora themselves directly, and not through the teacher's filter. Ideally, this would require appropriate corpora that correspond to their level of knowledge and learning objectives, software applications, computer labs, language teachers and motivated students who have good computer skills.[14]

[13] Martin – Pesovár 1963.
[14] Szirmai 2005.

LITERATURE

Chomsky, Noam
1957: *Syntactic Structures.* The Hague: Mouton.
fügedi, János:
1995: Applying Dance Structure and Motive Collections by Computer – DanceStruct 1.0. *Proceedings of the 19th Biennial Conference of ICKL,* Paris, 71–82.
2008: Unit Timing of Touching Gestures. *Proceedings of the 25th Biennial Conference of ICKL, Mexico City*, 33–48.
Hunston, Susan
2002: *Corpora in Applied Linguistics*. Cambridge: Cambridge University Press.
Hutchinson, Ann
1977: *Labanotation. The System of Analyzing and Recording Movement.* Third edition, Revised. New York: Theatre Arts Books, London: Dance Books.
Knust, Albrecht
1979: *A Dictionary of Kinetography Laban.* Vols 1–2. London: Macdonald and Evans Ltd.
Martin, György – Pesovár, Ernő:
1963: Determination of Motive Type in Dance Folklore. *Acta Ethnographica Hungarica* 13, 295–331.
Misi, Gábor
2002: Labanatory. A Computer Program to Analyse Dance. *Booklet of Abstracts for the 22nd Symposium of the ICTM Study Group on Ethnochoreology,* Szeged, 18.
2005: *Az erdélyi férfitáncok formális elemzéséről. A tévesztések, korrigálások, illetve általában a táncelem-kapcsolatok vizsgálatával.* [A Formal Analysis of Transylvanian Male Solo Dances, Including Mistakes, Corrections and Connections in Improvisation.] Manuscript. MTA ZTI Akt. 1653. Also available online: http://www.labanatory.com/common/download/2006MGYE/Formalis.pdf
2008: An Algebraic Representation of Labanotation for Retrieval and Other Operations. *Proceedings of the 25th Biennial Conference of ICKL, Mexico City,* 143–160.
2012: Formal Methods in Form Analysis of Transylvanian Male Solo Dances. In: Ivancich Dunin, Elsie – Giurchescu, Anca – Könczei, Csilla (eds): *From Field to Text & Dance and Space. Proceedings for the 24th Symposium of the ICTM Study Group on Ethnochoreology, Cluj Napoca,* 47–57.
Szentpál, Mária:
[1976]: *Táncjelírás. Laban-kinetográfia.* I., II., III. kötet. [Dance Notation. Laban Kinetography. Vols 1, 2 and 3.] Budapest: Népművelési Propaganda Iroda, 1976, 1969, 1973.
Szirmai, Monika
2005: *Bevezetés a korpusznyelvészetbe. A korpusznyelvészet alkalmazása az anyanyelv és az idegen nyelv tanulásában és tanításában.* [An Introduction to Corpus Linguistics. Applying Corpus Linguistics in the Learning and Teaching of Native and Foreign Languages.] Budapest: Tinta Könyvkiadó.

APPENDIX

Menus

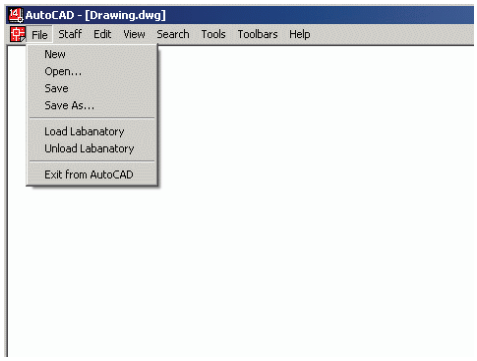


Figure 1. File menu

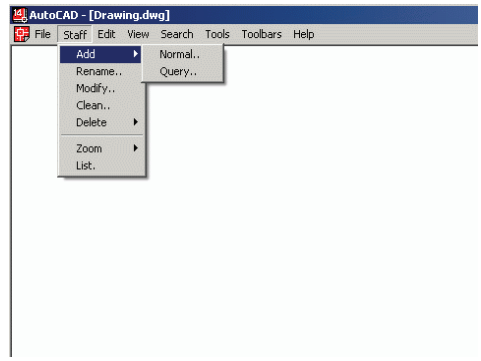


Figure 2. Staff menu

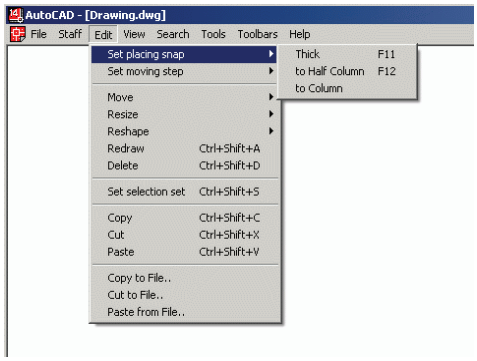


Figure 3. Edit menu

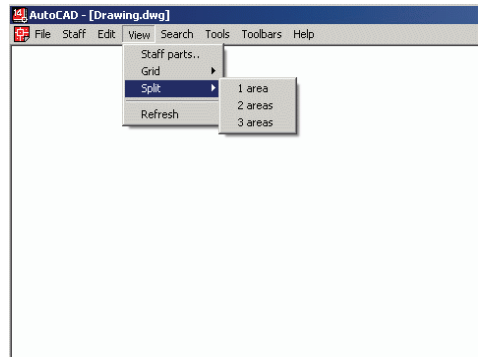


Figure 4. View menu

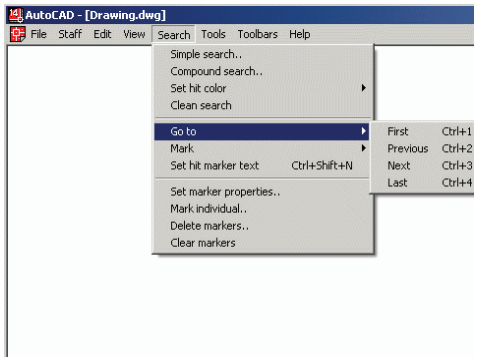


Figure 5. Search menu

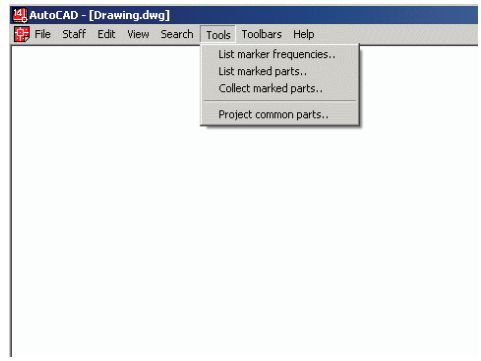


Figure 6. Tools menu

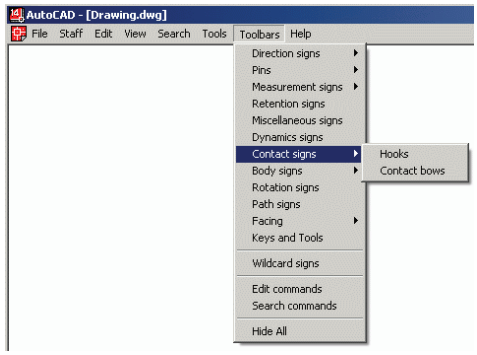


Figure 7. Toolbars menu

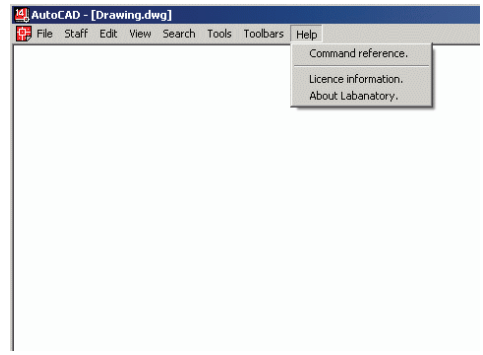


Figure 8. Help menu

Toolbars

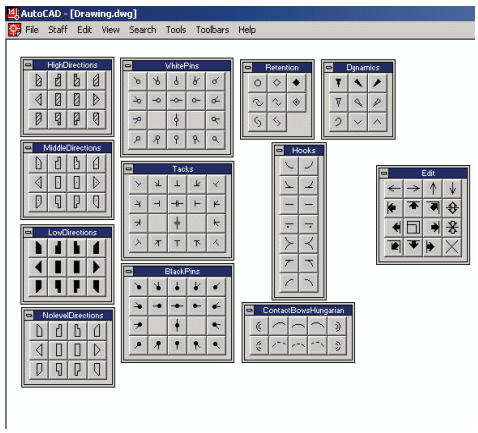


Figure 9. Icons of Labanotation signs and Edit

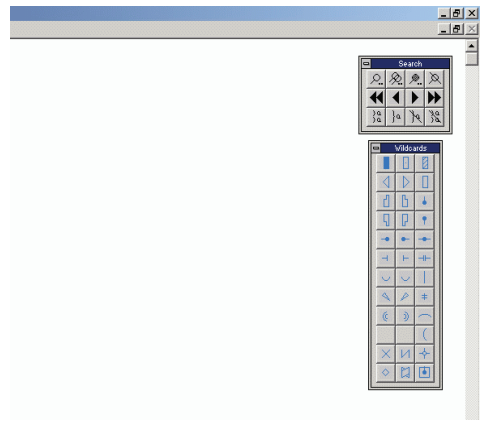


Figure 10. Icons of wildcard signs and Search

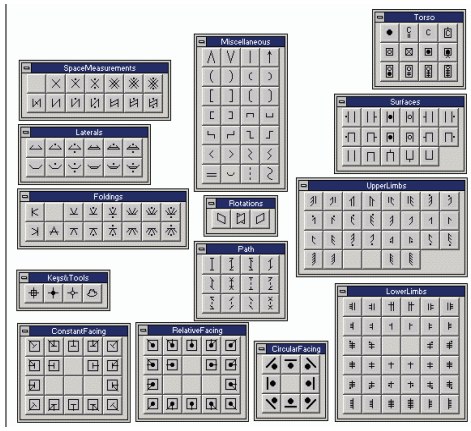


Figure 11. Icons of additional Labanotation signs

Command reference

```
-------------------------------------------------------------------------
 Format:      'refresh'
 Description: refreshes all parts of the open drawing
-------------------------------------------------------------------------
 Format:      'split _ view' 1|2|3
 Description: splits the view into more parts
-------------------------------------------------------------------------
 Format:      'new _ file'
 Description: drops the open file and opens a new empty one
-------------------------------------------------------------------------
 Format:      'load _ file'
 Description: drops the open file and opens an other with user dialog
-------------------------------------------------------------------------
 Format:      'add _ staff' NORMAL|QUERY <staff _ name>
              <beats _ per _ starting _ pos> <beats _ per _ upbeat>
              <beats _ per _ measure> <measures _ per _ staff>
              <numbering _ distance _ from _ left>
 Description: adds a staff to the current drawing
-------------------------------------------------------------------------
 Format:      'rename _ staff' <staff _ name> <staff _ new _ name>
 Description: renames a staff in the current drawing
-------------------------------------------------------------------------
 Format:      'modify _ staff' <staff _ name>
              <beats _ per _ starting _ pos> <beats _ per _ upbeat>
              <beats _ per _ measure> <measures _ per _ staff>
              <numbering _ distance _ from _ left>
 Description: modifies a staff in the current drawing
-------------------------------------------------------------------------
 Format:      'clean _ staff' <staff _ name> YES|NO
 Description: deletes the signs in a staff
-------------------------------------------------------------------------
 Format:      'delete _ staff' <staff _ name> YES|NO
 Description: deletes a staff from the current drawing
-------------------------------------------------------------------------
 Format:      'delete _ in _ cycle' NORMAL|QUERY|ANY <name _ mask>
 Description: deletes staves from the current drawing in sequence
-------------------------------------------------------------------------
 Format:      'zoom _ to _ staff' <staff _ name> <measure _ no>
              <visible _ number _ of _ beats>
 Description: zooms a staff measure
-------------------------------------------------------------------------
 Format:      'zoom _ in _ cycle' NORMAL|QUERY|ANY <name _ mask>
              <visible _ number _ of _ beats>
 Description: zooms staves in sequence
```

```
-------------------------------------------------------------------------
 Format:      'list _ staff _ names'
 Description: lists the names of all the registered staves
-------------------------------------------------------------------------
 Format:      'show _ staff _ parts'
              <is _ visible _ support _ line> <is _ visible _ gesture _ lines>
              <is _ visible _ measure _ lines> <is _ visible _ tick _ marks>
              <is _ visible _ staff _ names> <is _ visible _ measure _ numbers>
              <is _ visible _ beat _ numbers> <is _ visible _ bookmarks>
 Description: shows or hides parts of all the staves
-------------------------------------------------------------------------
 Format:      'place _ sign' <basic _ block _ name>
 Description: places a sign with user interaction
-------------------------------------------------------------------------
 Format:      'set _ selection _ set'
 Description: sets a sign set by user selection e.g. to move signs
-------------------------------------------------------------------------
 Format:      'delete _ sign'
 Description: deletes signs of the selection set
-------------------------------------------------------------------------
 Format:      'move _ sign' UP|DOWN|LEFT|RIGHT
 Description: moves signs of the selection set
-------------------------------------------------------------------------
 Format:      'resize _ sign' INCREASE|DECREASE|NORMALIZE
              UP|DOWN|LEFT|RIGHT|LEFTUP|RIGHTUP
 Description: resizes the only sign of the selection set
-------------------------------------------------------------------------
 Format:      'reshape _ sign' OUT|IN
 Description: reshapes the only sign of the selection set
-------------------------------------------------------------------------
 Format:      'redraw _ sign'
 Description: redraws signs of the selection set
-------------------------------------------------------------------------
 Format:      'cut _ sign' <file _ name>
 Description: cuts the selection set to a file
-------------------------------------------------------------------------
 Format:      'copy _ sign' <file _ name>
 Description: copies the selection set to a file
-------------------------------------------------------------------------
 Format:      'paste _ sign' <file _ name>
 Description: pastes a saved drawing part from a file
```

```
-------------------------------------------------------------------------
 Format:      'simple _ search' <query _ staff _ name> <normal _ staff _ name>
              ORIGINAL|AUGMENTED|BOTH  IDENTICAL|SYMMETRICAL|BOTH
              MEASURE|HALFMEASURE|NONMEASURE|AMONGHALFS|ANY
              LENGTHANDPOSITION|ONLYPOSITION
 Description: searches the pattern of query staff in a normal one
-------------------------------------------------------------------------
 Format:      'compound _ search' {<OR _ query _ staff _ name>}
              {<NOT _ query _ staff _ name>} <normal _ staff _ name>
              ORIGINAL|AUGMENTED|BOTH  IDENTICAL|SYMMETRICAL|BOTH
              MEASURE|HALFMEASURE|NONMEASURE|AMONGHALFS|ANY
              LENGTHANDPOSITION|ONLYPOSITION
 Description: searches the logical expression of staves in a normal staff
-------------------------------------------------------------------------
 Format:      'clean _ search'
 Description: clears the result of a previous search
-------------------------------------------------------------------------
 Format:      'set _ hit _ color' <color _ code>
 Description: sets the color of found elements
-------------------------------------------------------------------------
 Format:      'go _ to _ hit' FIRST|PREVIOUS|NEXT|LAST
 Description: shows a found subpattern zooming it
-------------------------------------------------------------------------
 Format:      'mark _ hit' CURRENT|ALL ON|OFF
 Description: marks or unmarks found subpatterns
-------------------------------------------------------------------------
 Format:      'set _ hit _ marker _ text' <marker _ text>
 Description: sets text of markers for mark _ hit command
-------------------------------------------------------------------------
 Format:        'set _ marker _ properties' CONTINUOUS|DASHED <marker _ dis-
tance>
 Description: sets line type and distance of markers
-------------------------------------------------------------------------
 Format:      'mark _ individual' <staff _ name> <measure _ number>
              <grid _ shift> <grid _ length> <marker _ text>
 Description: marks an individual part of a staff
-------------------------------------------------------------------------
 Format:      'delete _ markers' <staff _ name> <marker _ text>
 Description: deletes markers in a staff
-------------------------------------------------------------------------
 Format:      'clear _ markers' YES|NO
 Description: deletes all the markers in the current drawing
```

```
-------------------------------------------------------------------------
 Format:      'list _ marker _ frequencies' <staff _ name>
 Description: counts certain markers from a staff
-------------------------------------------------------------------------
 Format:      'list _ marked _ parts' <staff _ name> <marker _ text>
 Description: lists places of certain marked parts from a staff
-------------------------------------------------------------------------
 Format:      'collect _ marked _ parts' <staff _ name> <marker _ text>
              <number _ of _ pre _ grids>
 Description: collects certain marked parts from a staff
-------------------------------------------------------------------------
 Format:      'project _ common _ parts' {<operand _ staff _ mask>}
 Description: projects the common sign parts of certain staves
-------------------------------------------------------------------------
 Format:      'labanatory _ reference'
 Description: writes this quick command reference
-------------------------------------------------------------------------
 Format:      'labanatory _ licence _ info'
 Description: writes the licence information
-------------------------------------------------------------------------
 Format:      'labanatory _ version'
 Description: lists the version of the program
-------------------------------------------------------------------------
```

Hotkeys

```
 [Ctrl+Shift+Z]    zoom _ in _ cycle
 [F11]             snap;0.25
 [F12]             snap;1
 [Ctrl+UpArrow]    move _ sign;up;
 [Ctrl+DownArrow]  move _ sign;down;
 [Ctrl+LeftArrow]  move _ sign;left;
 [Ctrl+RightArrow] move _ sign;right;
 [Ctrl+Numeric8]   resize _ sign;increase;up;
 [Ctrl+Numeric7]   resize _ sign;increase;left;
 [Ctrl+Numeric6]   resize _ sign;increase;right;
 [Ctrl+Numeric9]   resize _ sign;increase;rightup;
 [Ctrl+Numeric5]   resize _ sign;normalize;rightup;
 [Ctrl+Numeric2]   resize _ sign;decrease;up;
 [Ctrl+Numeric3]   resize _ sign;decrease;left;
 [Ctrl+Numeric4]   resize _ sign;decrease;right;
 [Ctrl+Numeric1]   resize _ sign;decrease;rightup;
 [Ctrl+PageUp]     reshape _ sign;out;
```

```
[Ctrl+PageDown]    reshape _ sign;in;
[Ctrl+Shift+A]     redraw _ sign;
[Ctrl+Shift+D]     delete _ sign;
[Ctrl+Shift+S]     set _ selection _ set
[Ctrl+Shift+C]     copy _ sign;clpbrd;
[Ctrl+Shift+X]     cut _ sign;clpbrd;
[Ctrl+Shift+V]     paste _ sign;clpbrd;
[Ctrl+1]           go _ to _ hit;first;
[Ctrl+2]           go _ to _ hit;previous;
[Ctrl+3]           go _ to _ hit;next;
[Ctrl+4]           go _ to _ hit;last;
[Ctrl+Shift+N]     set _ hit _ marker _ text;
```

Screenshots of executed command outputs

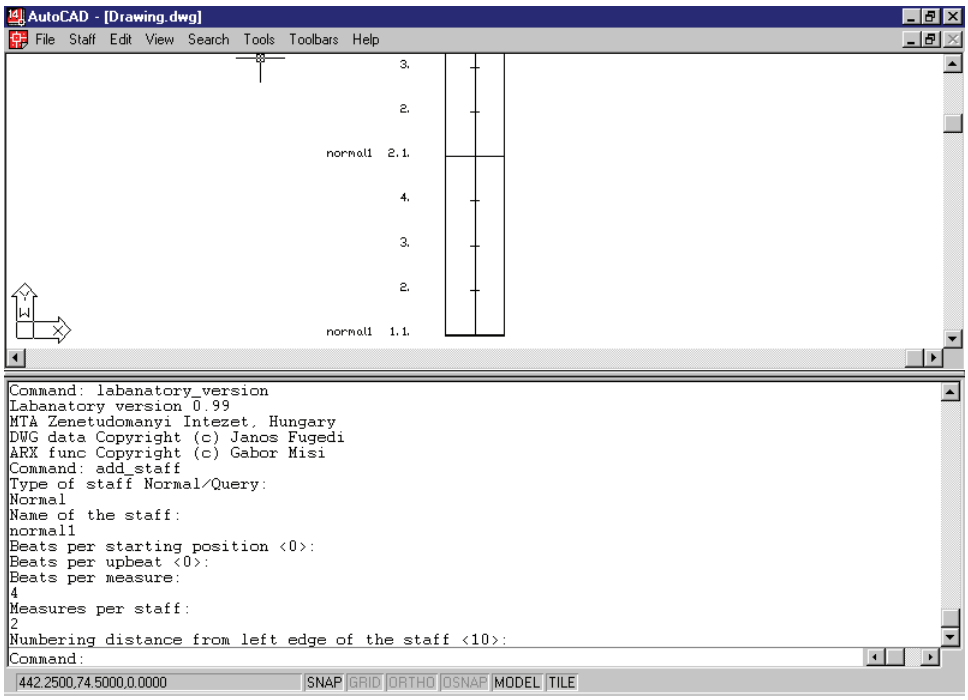


Figure 12. Adding a staff

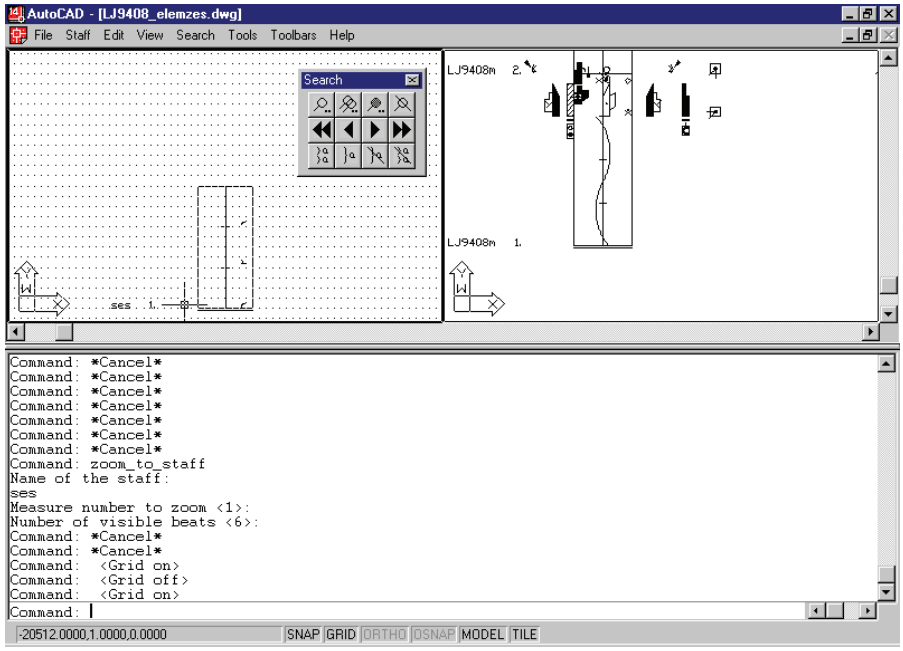


Figure 13. Making the search grid visible

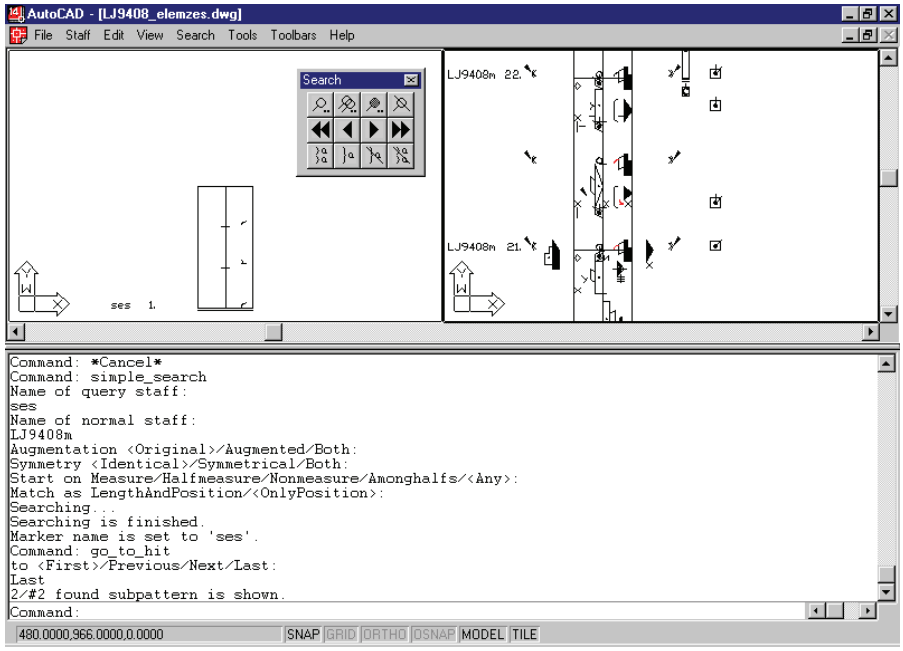


Figure 14. Simple search

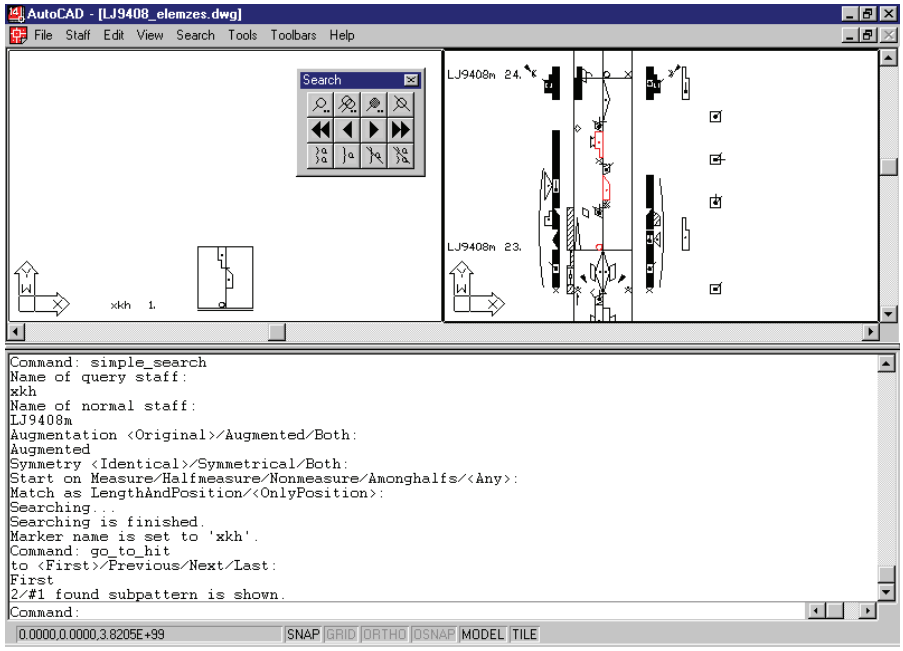


Figure 15. Augmented search

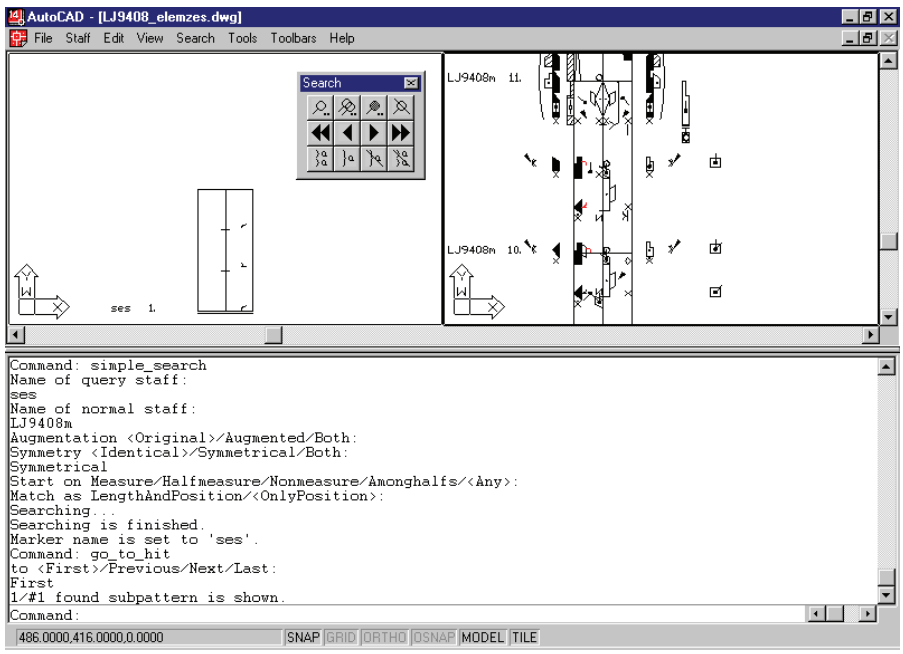


Figure 16. Symmetrical search

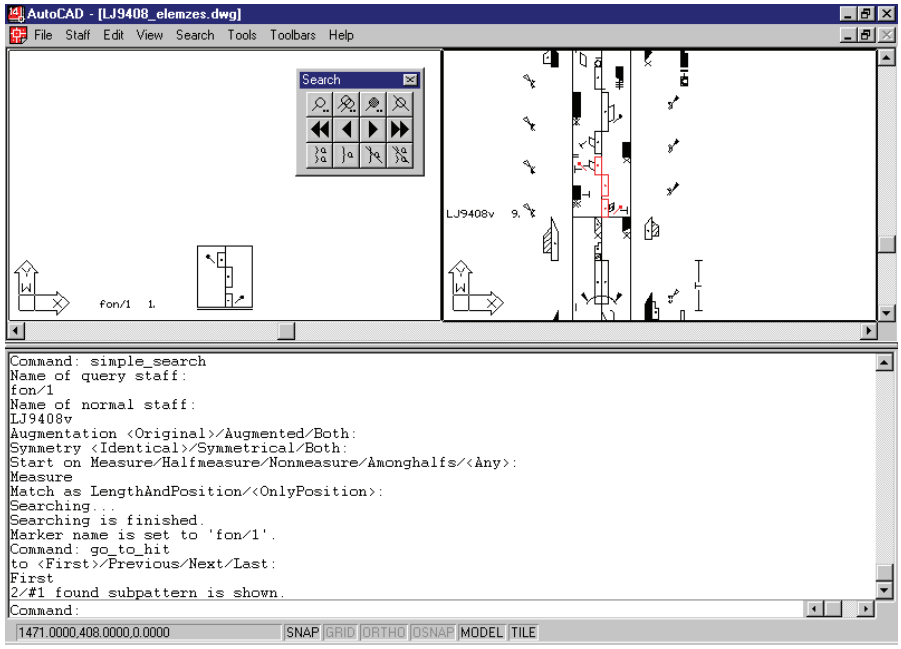


Figure 17. Search regarding measurement lines

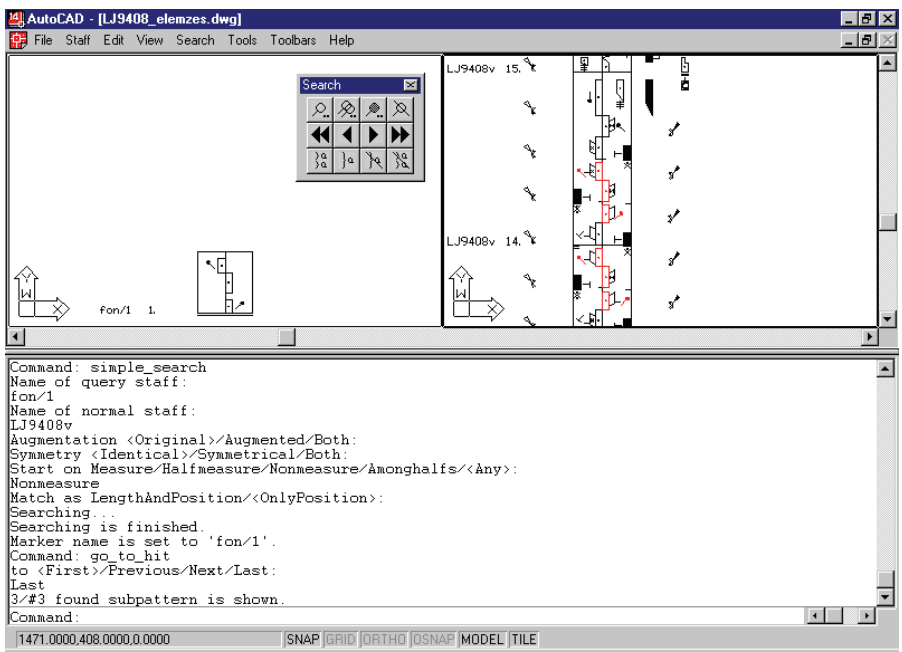


Figure 18. Search disregarding measurement lines

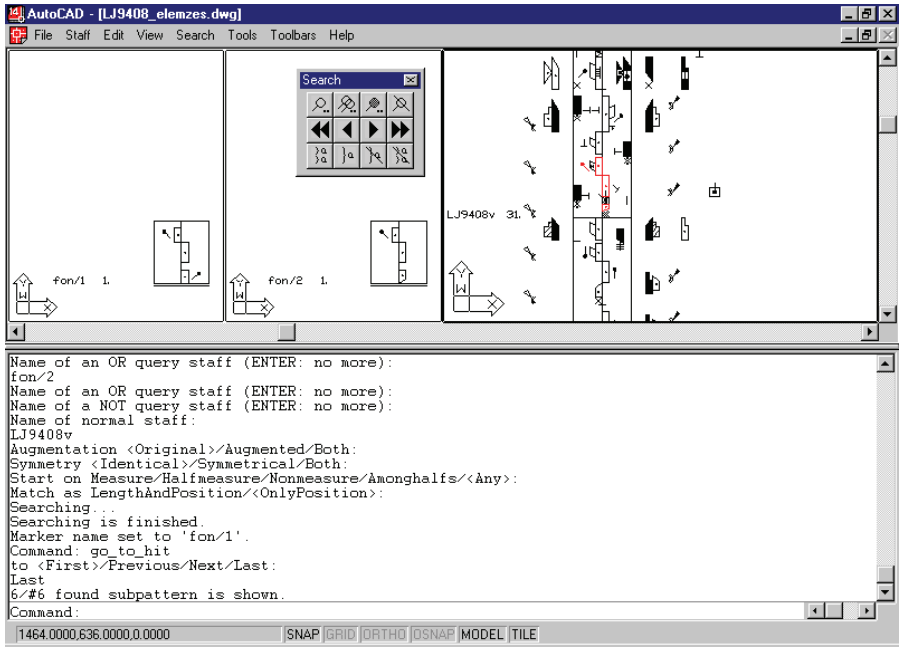


Figure 19. Compound OR search

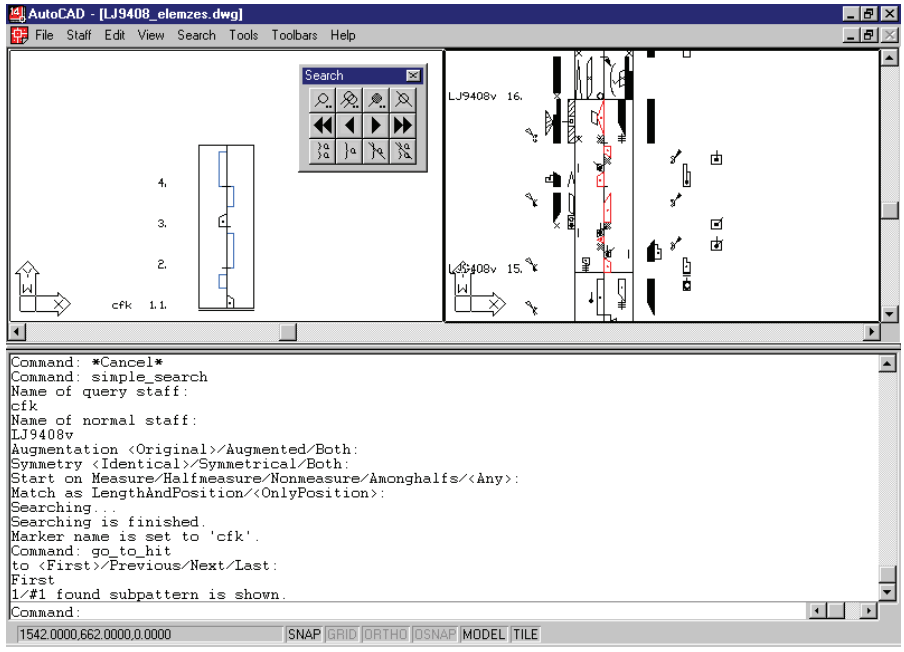


Figure 20. Search with wildcard signs

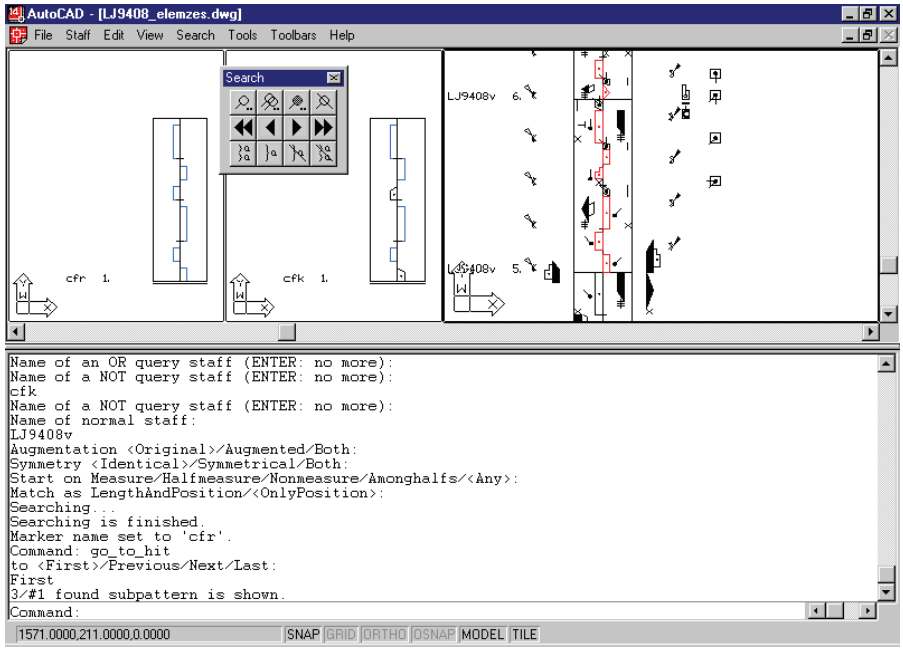


Figure 21. Compound NOT search with wildcard signs

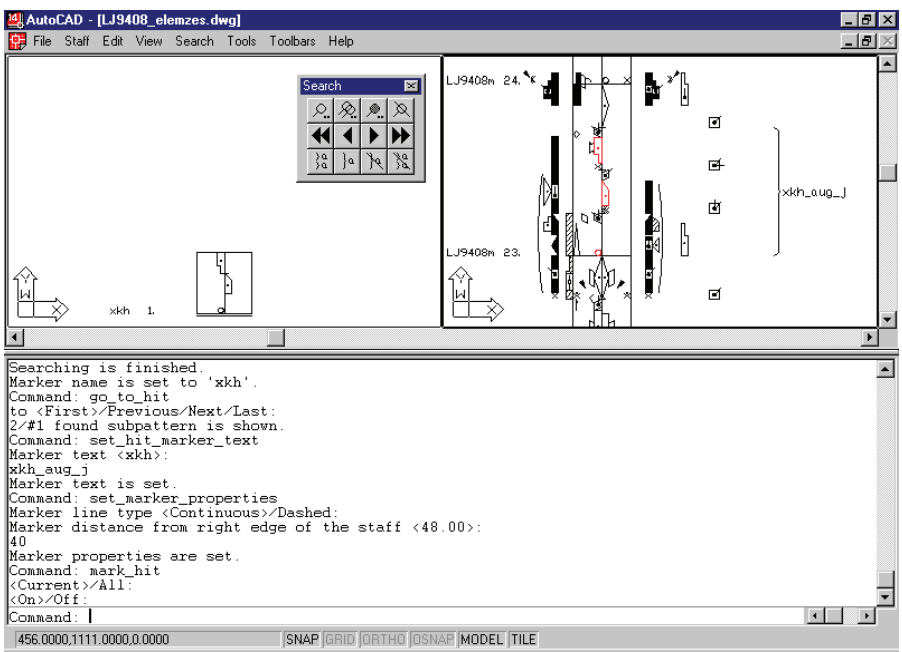


Figure 22. Marking a hit

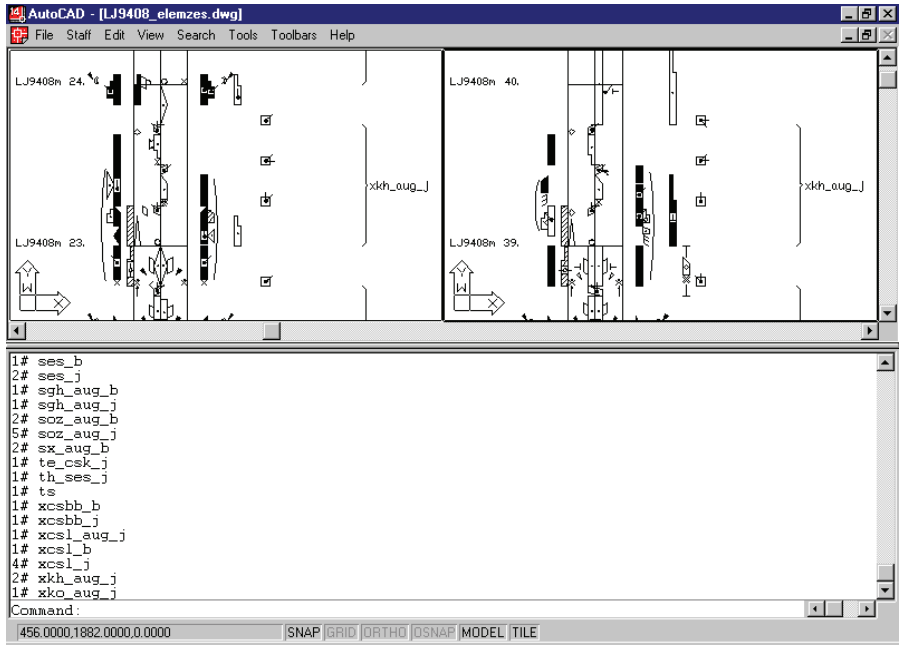


Figure 23. Listing markers and frequencies

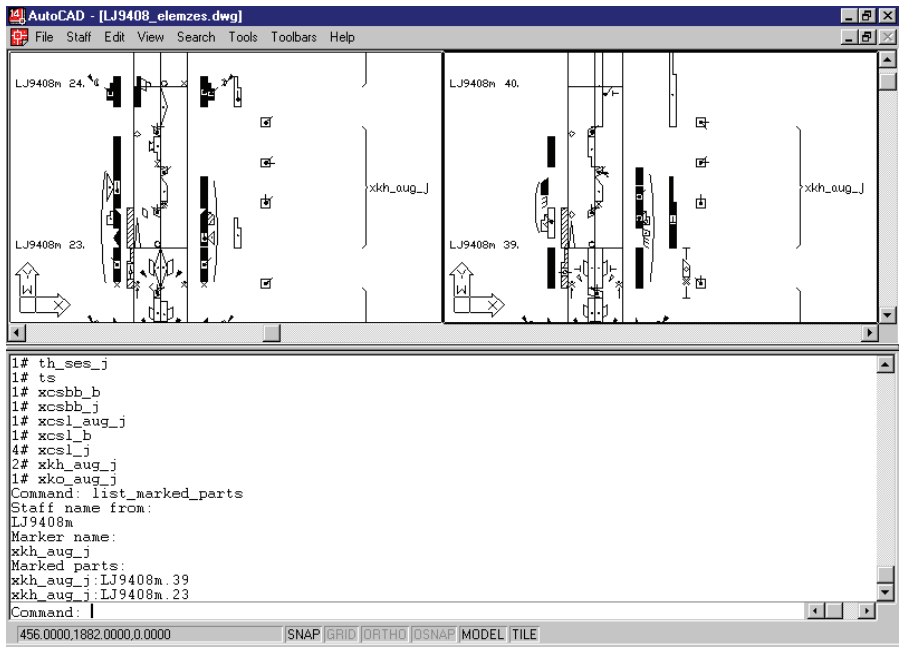


Figure 24. Listing marked parts

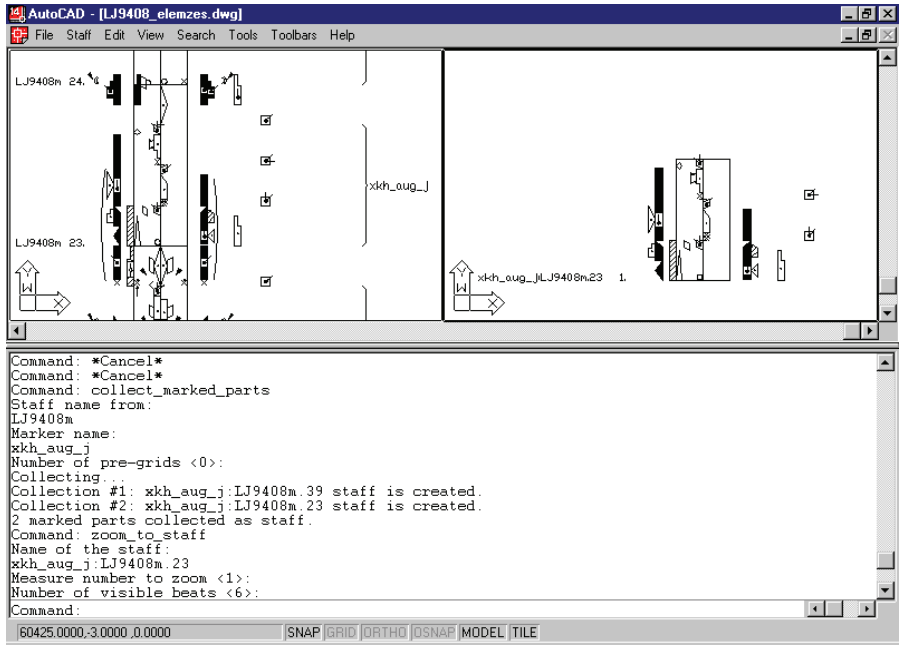


Figure 25. Creating a new staff with the content of a marked staff part

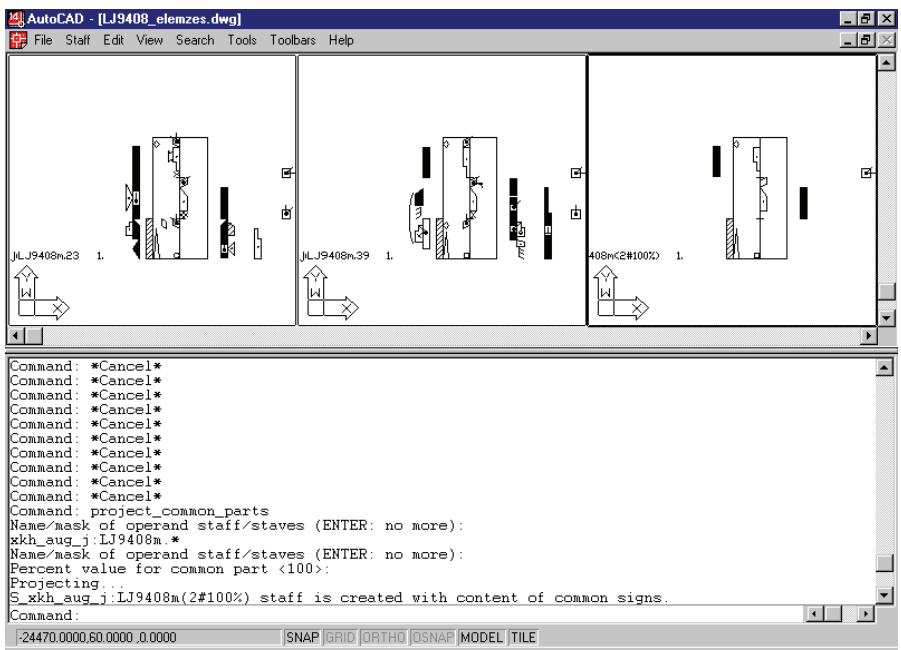


Figure 26. Determining the intersection of several sign patterns

Figure 27. A simplified corpus for analysis

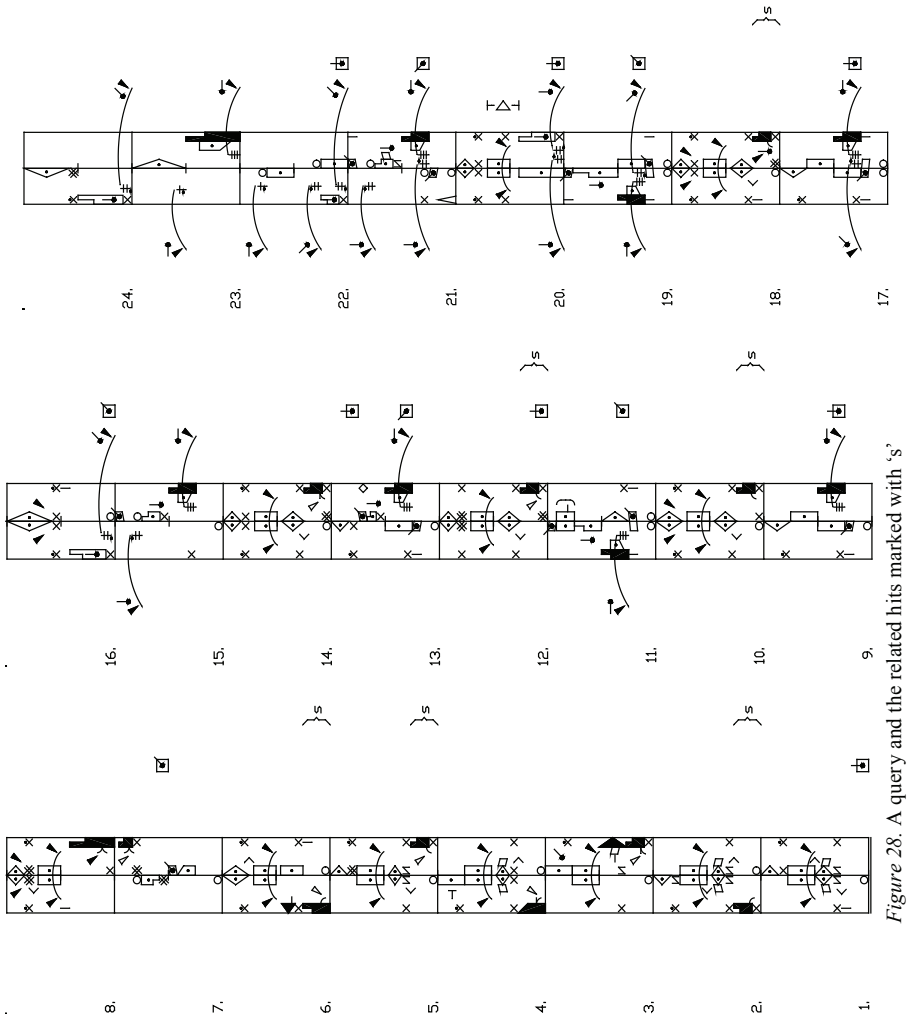


Figure 28. A query and the related hits marked with 's'

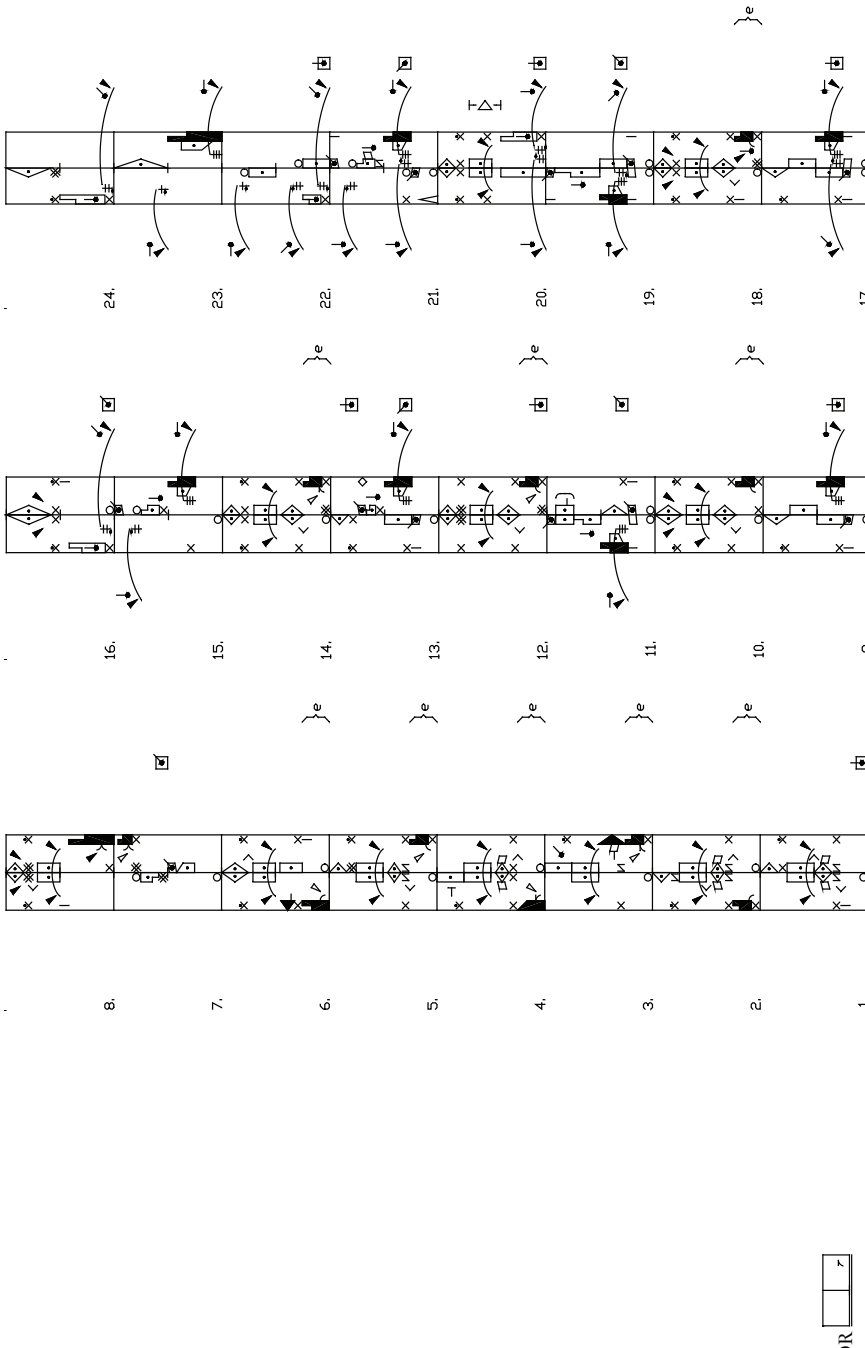


Figure 29. A query and the related hits marked with 'e'

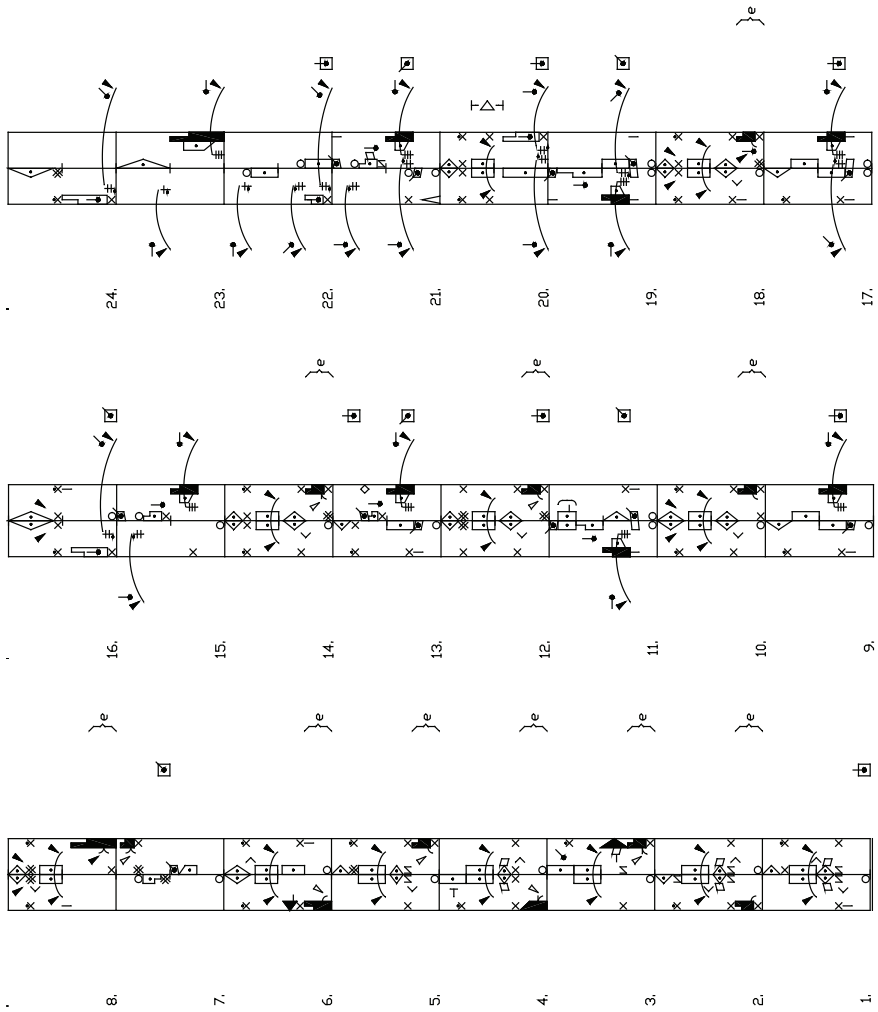


Figure 30. A query and the related hits marked with ‘e’

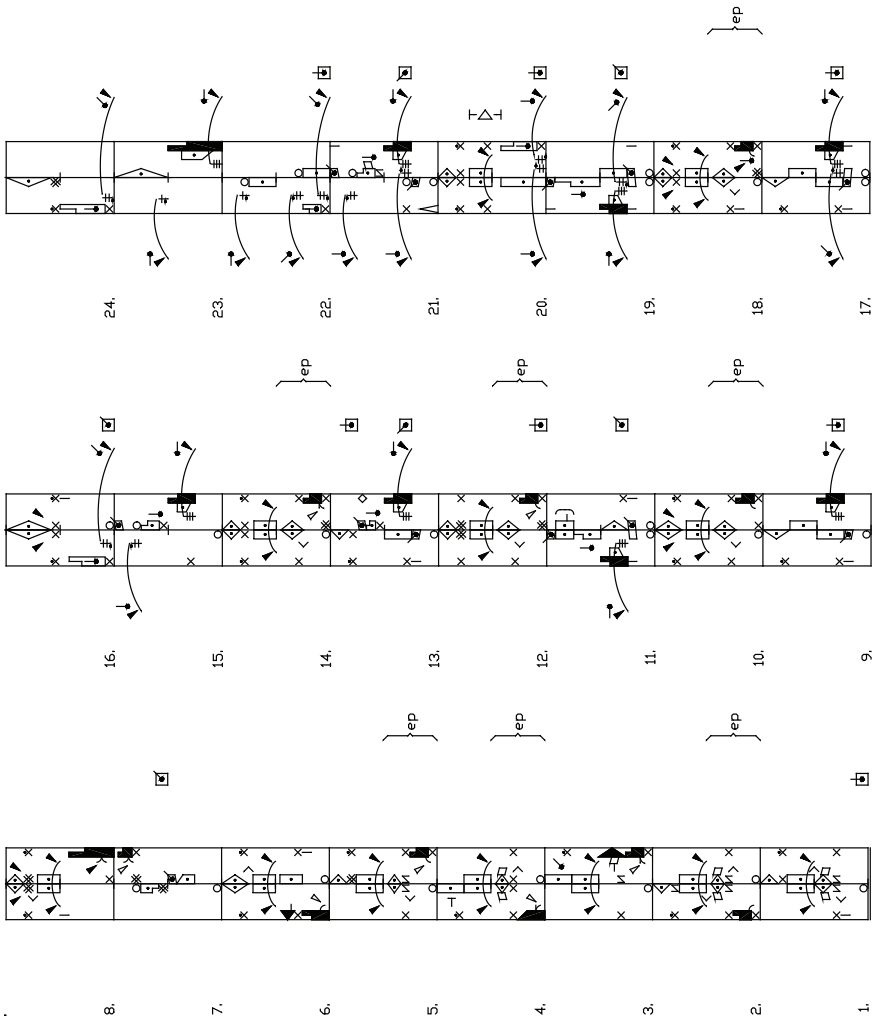


Figure 31. A query and the related hits marked with 'ep'

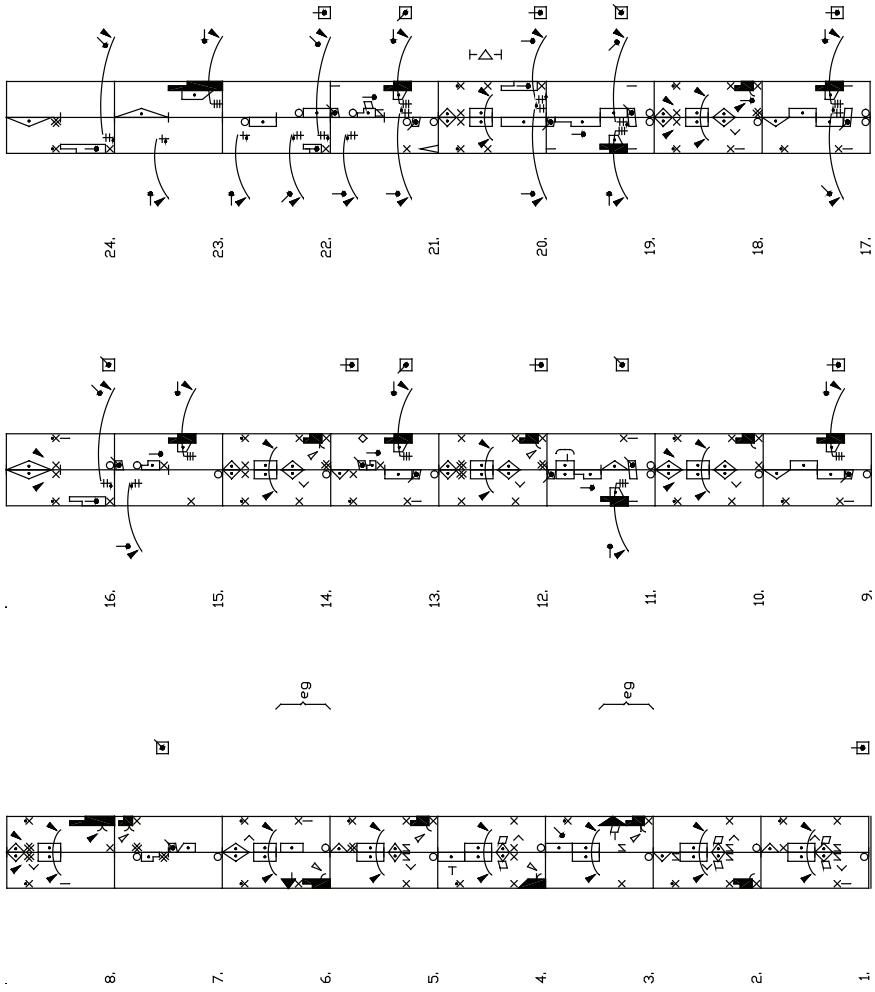


Figure 32. A query and the related hits marked with 'eg'

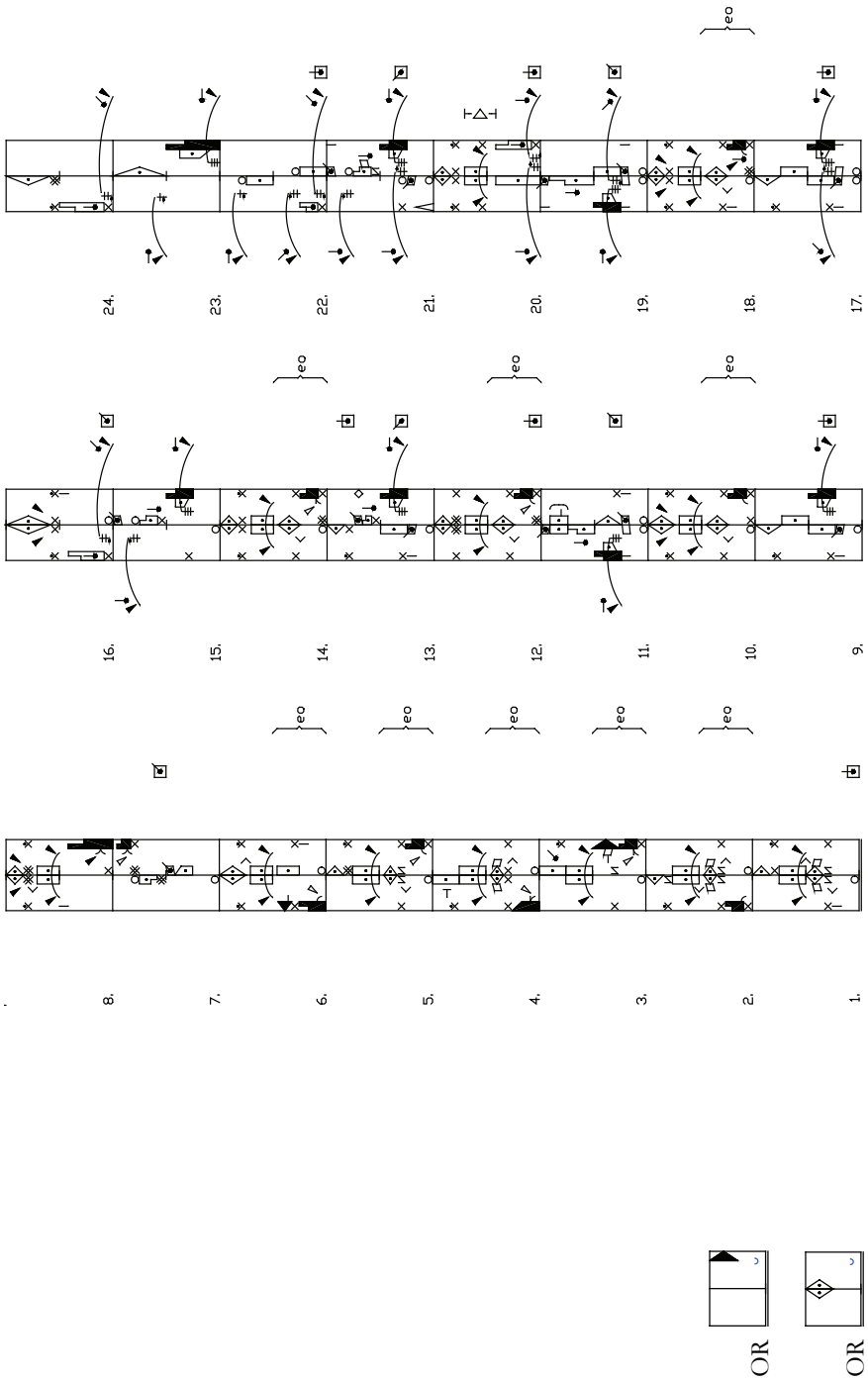


Figure 33. A query and the related hits marked with 'eo'

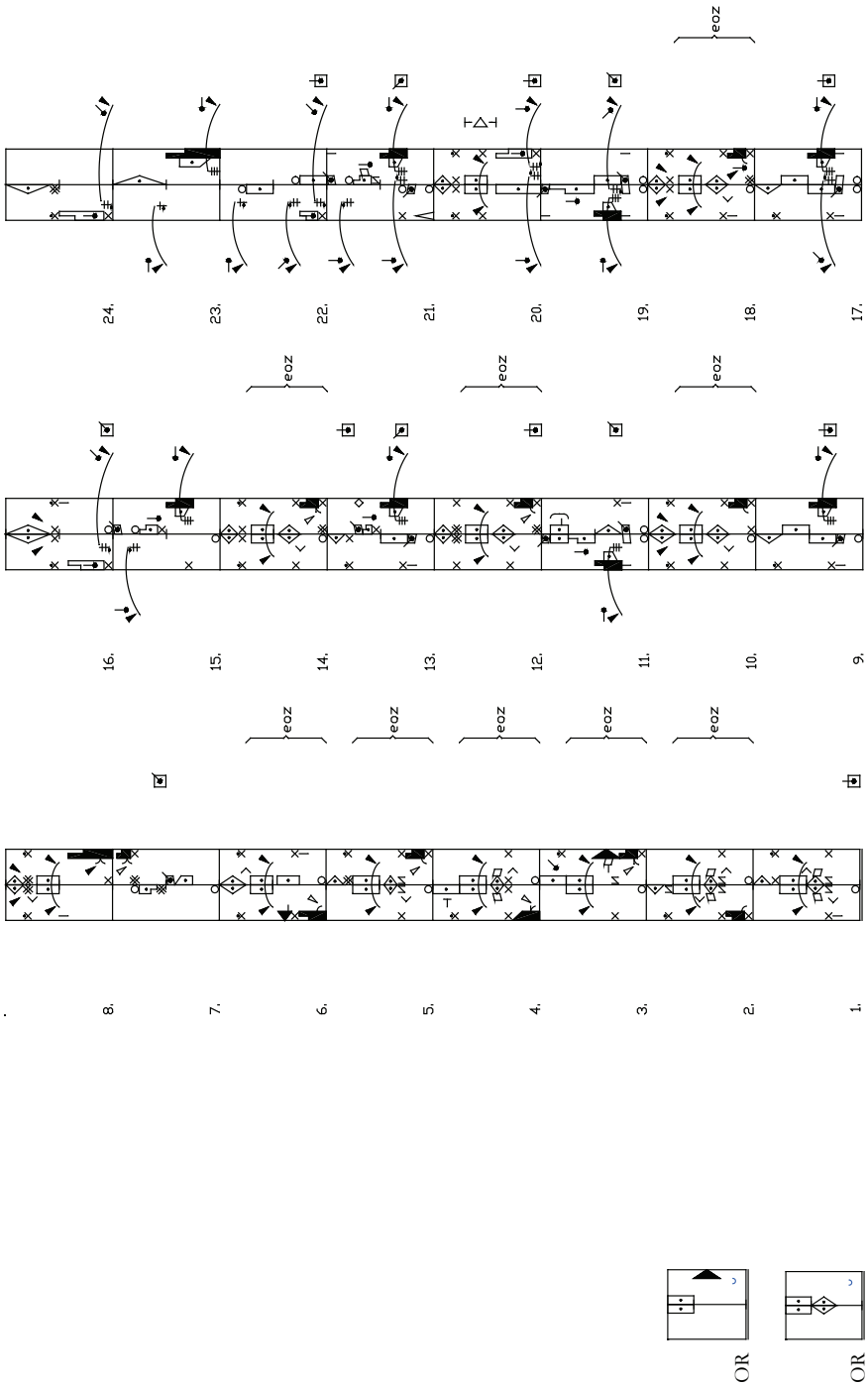


Figure 34. A query and the related hits marked with 'eoz'

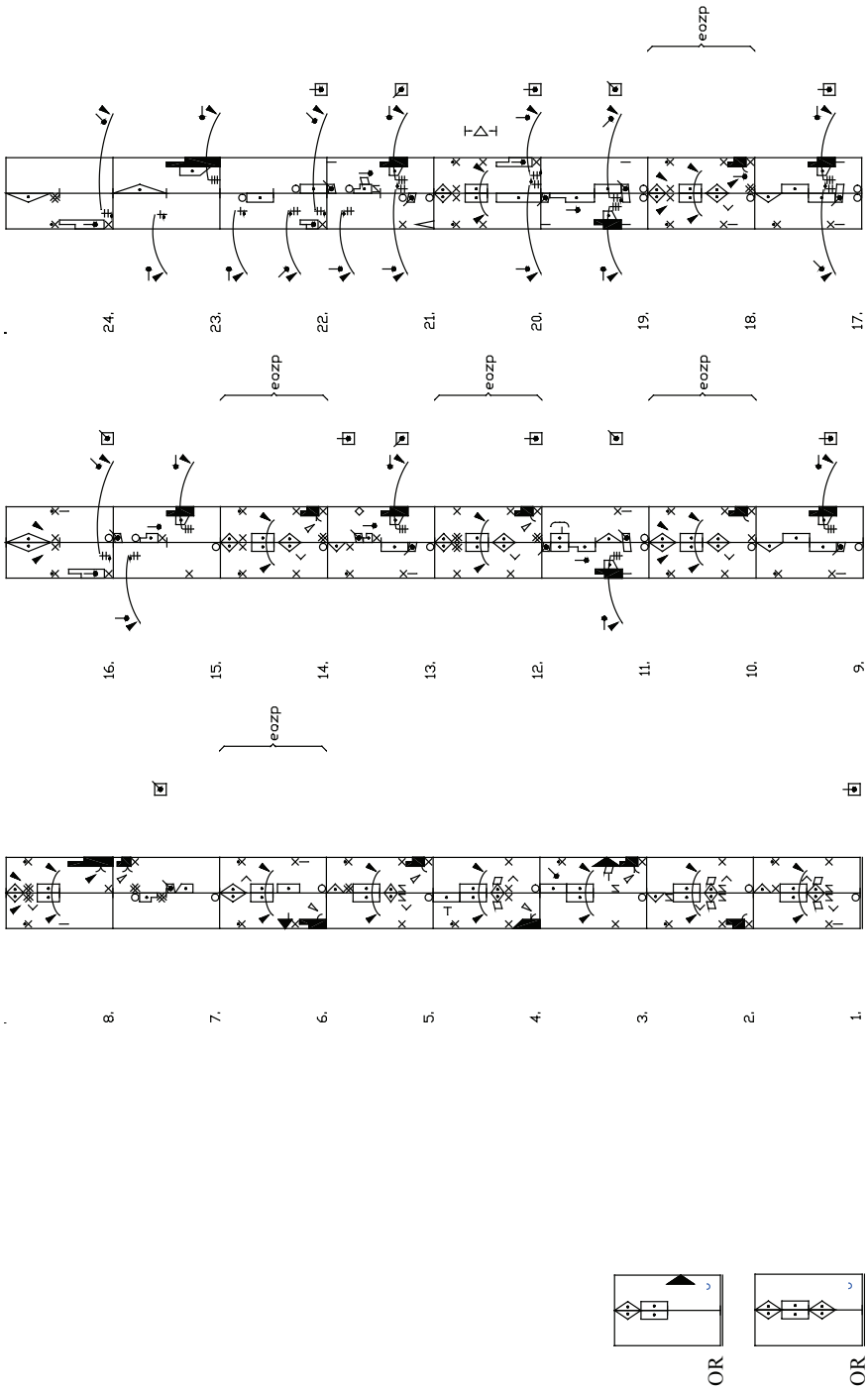


Figure 35. A query and the related hits marked with 'eozp'

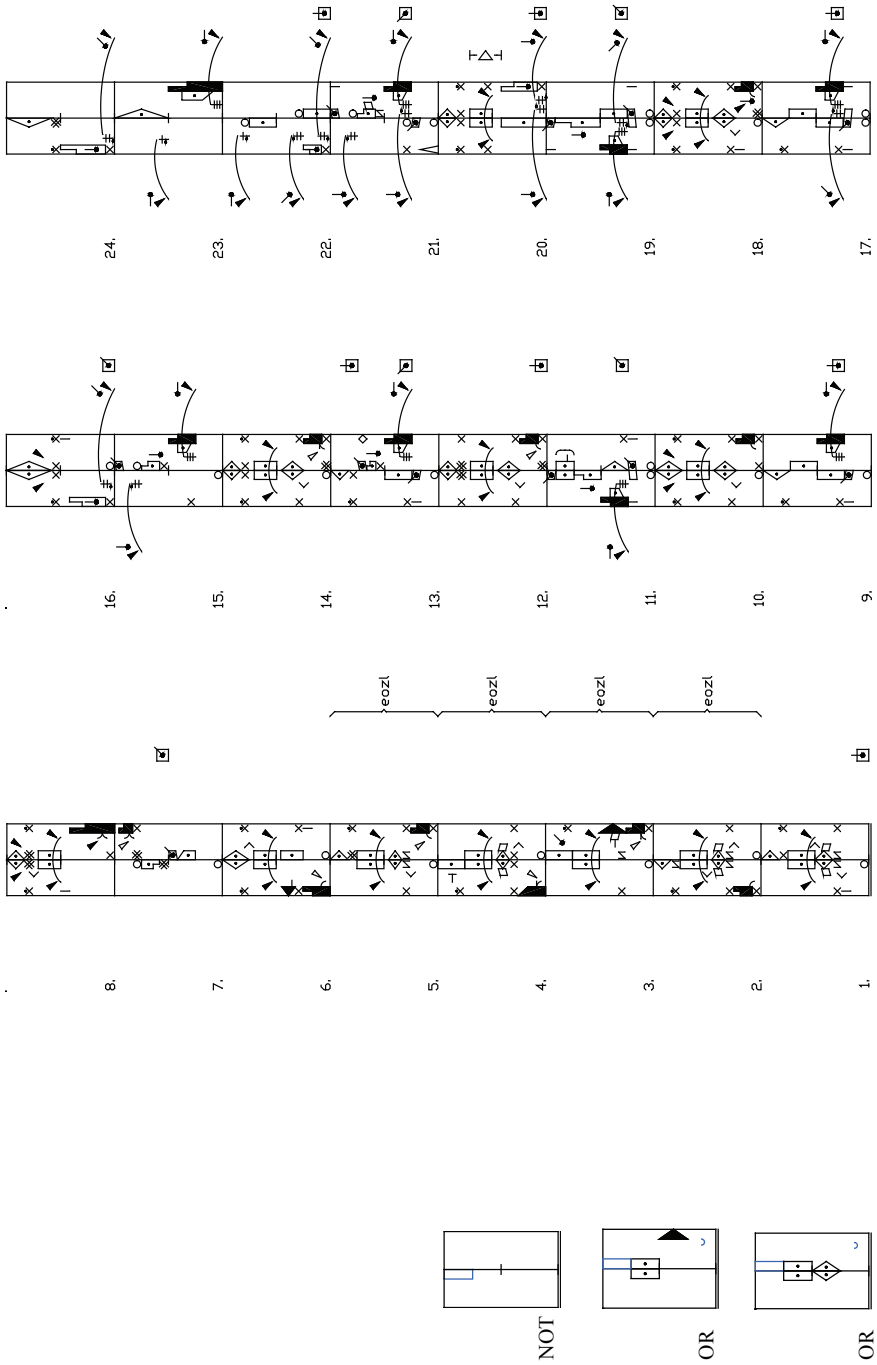


Figure 36. A query and the related hits marked with ‘eozl’

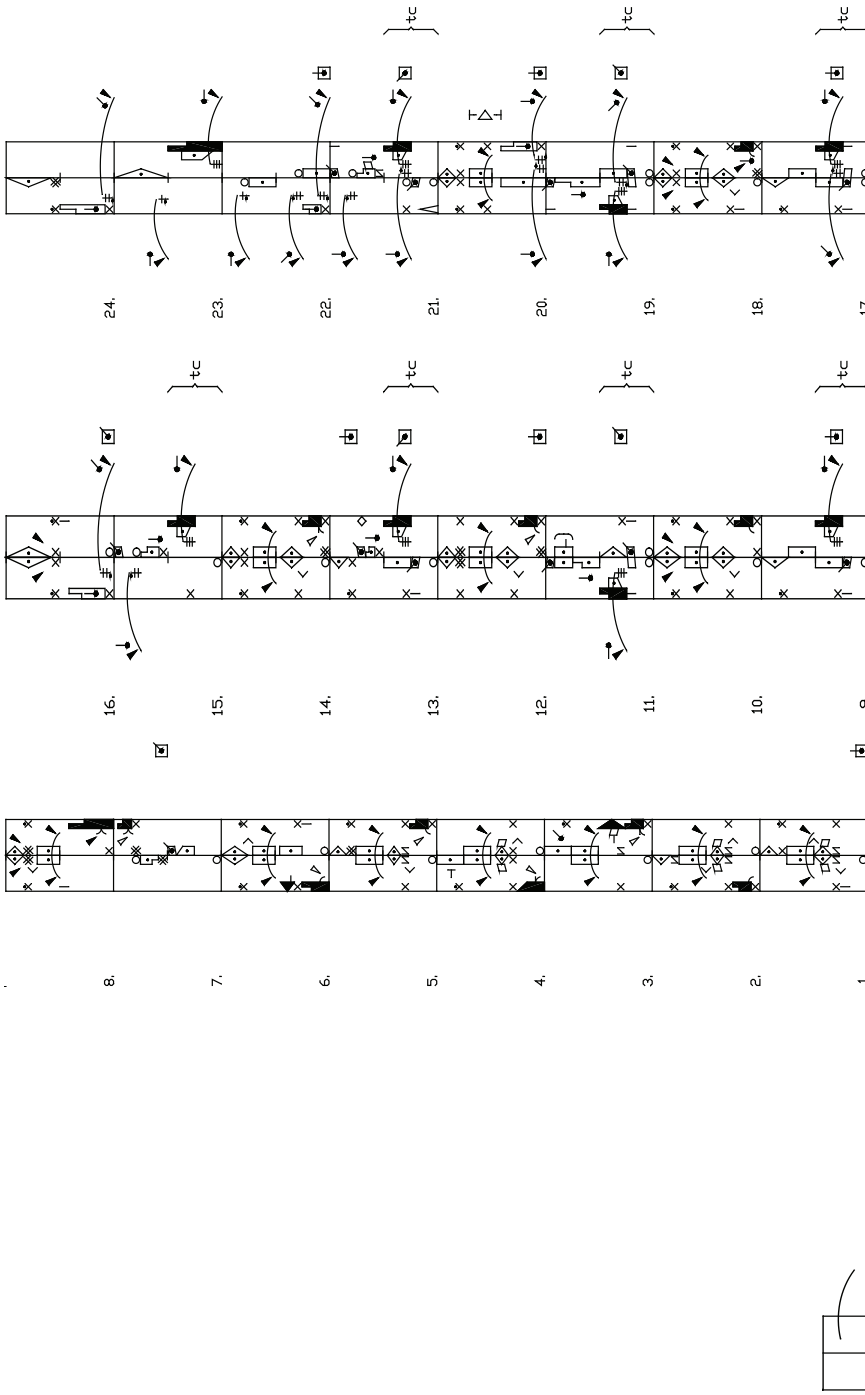


Figure 37. A query and the related hits marked with 'tc'

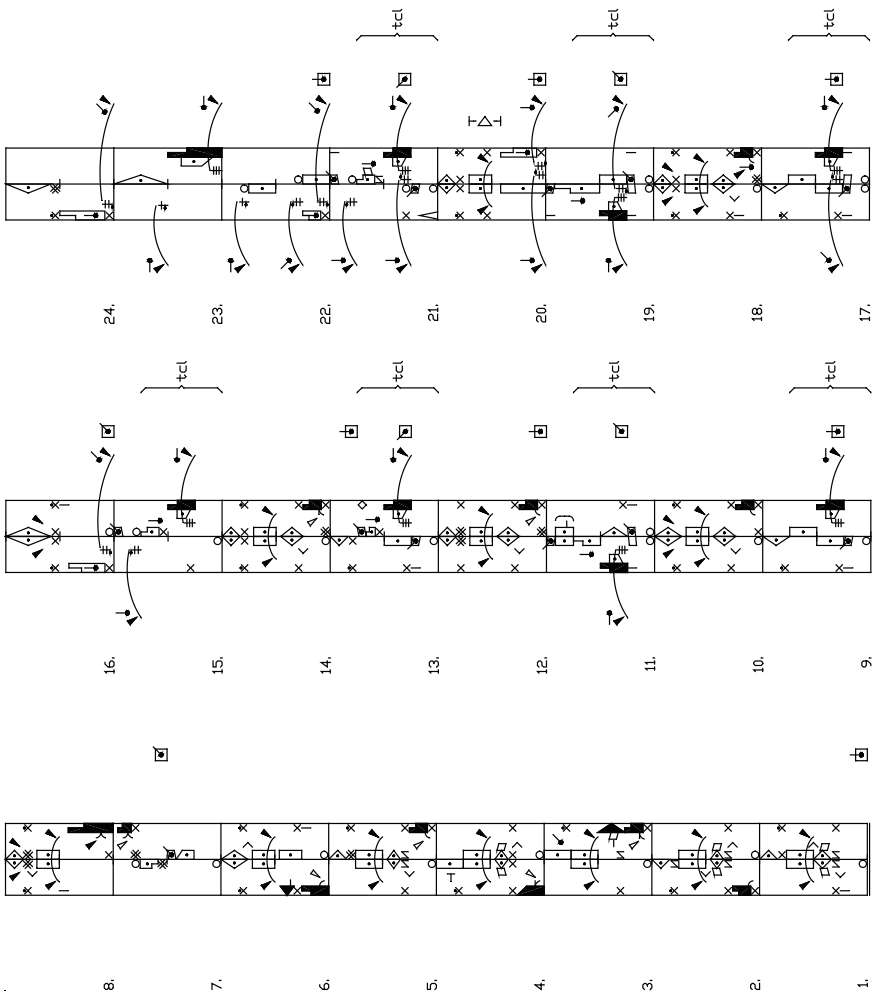
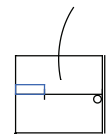


Figure 38. A query and the related hits marked with 'tcl'



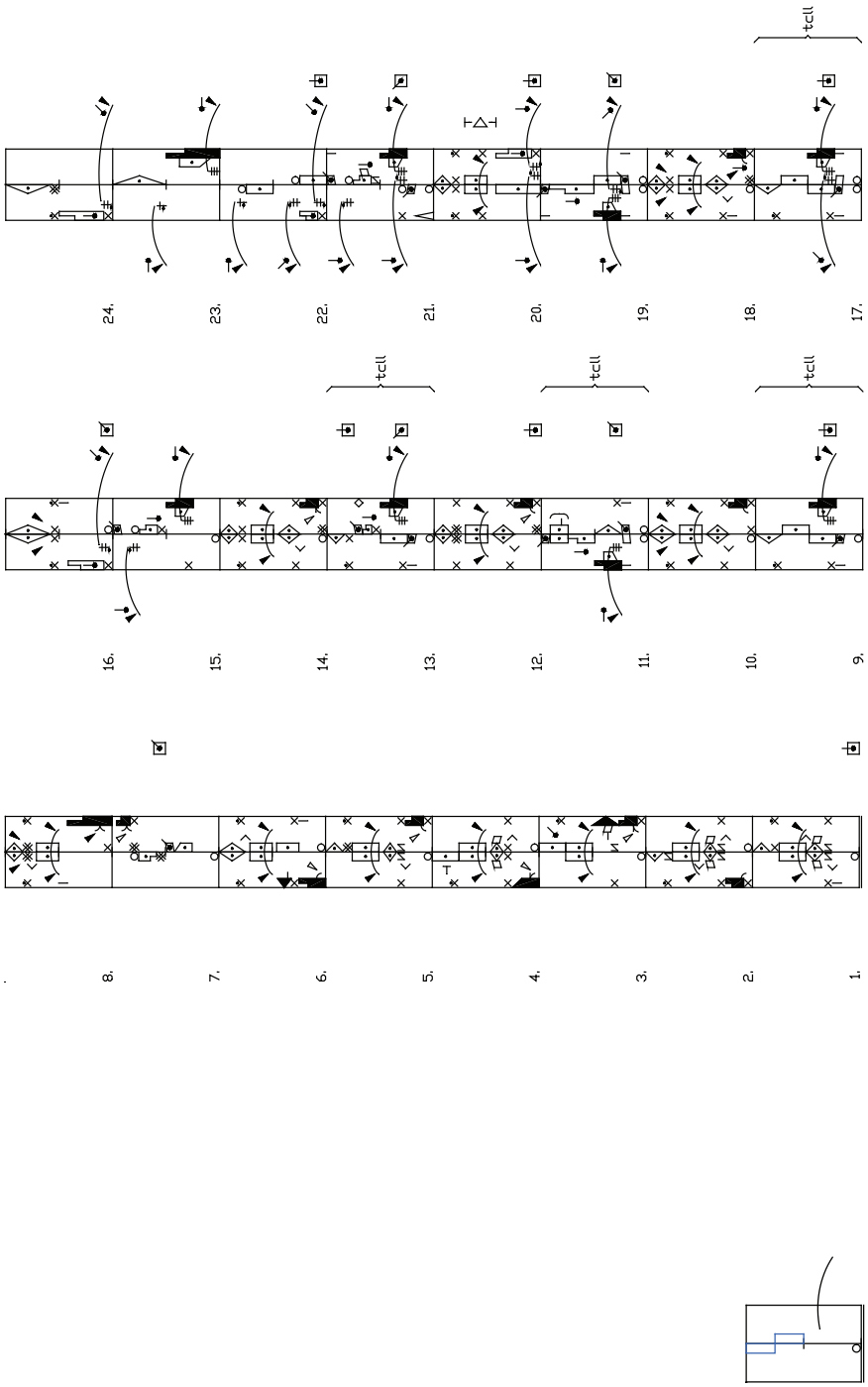


Figure 39. A query and the related hits marked with 'tcll'

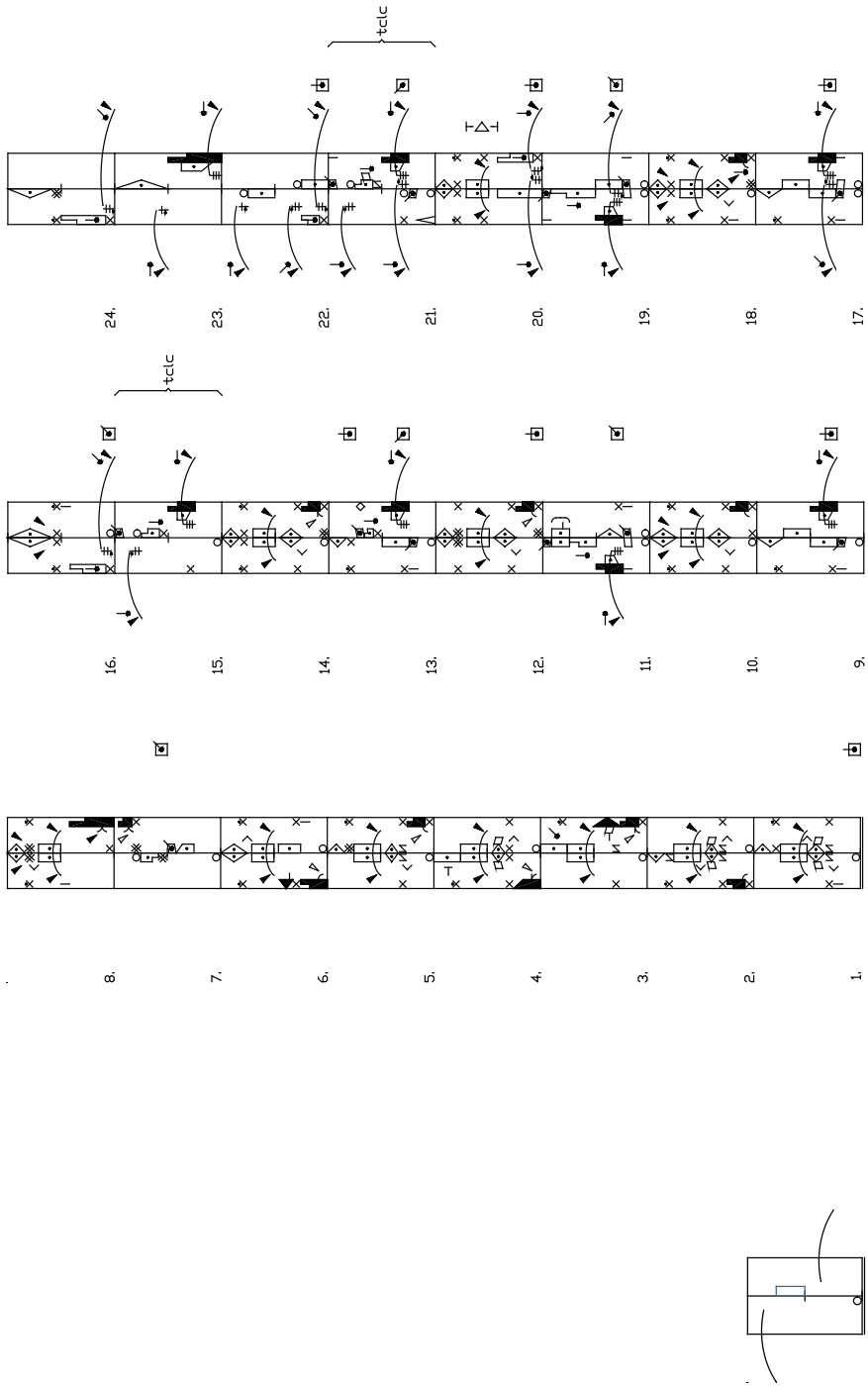


Figure 40. A query and the related hits marked with 'tclc'

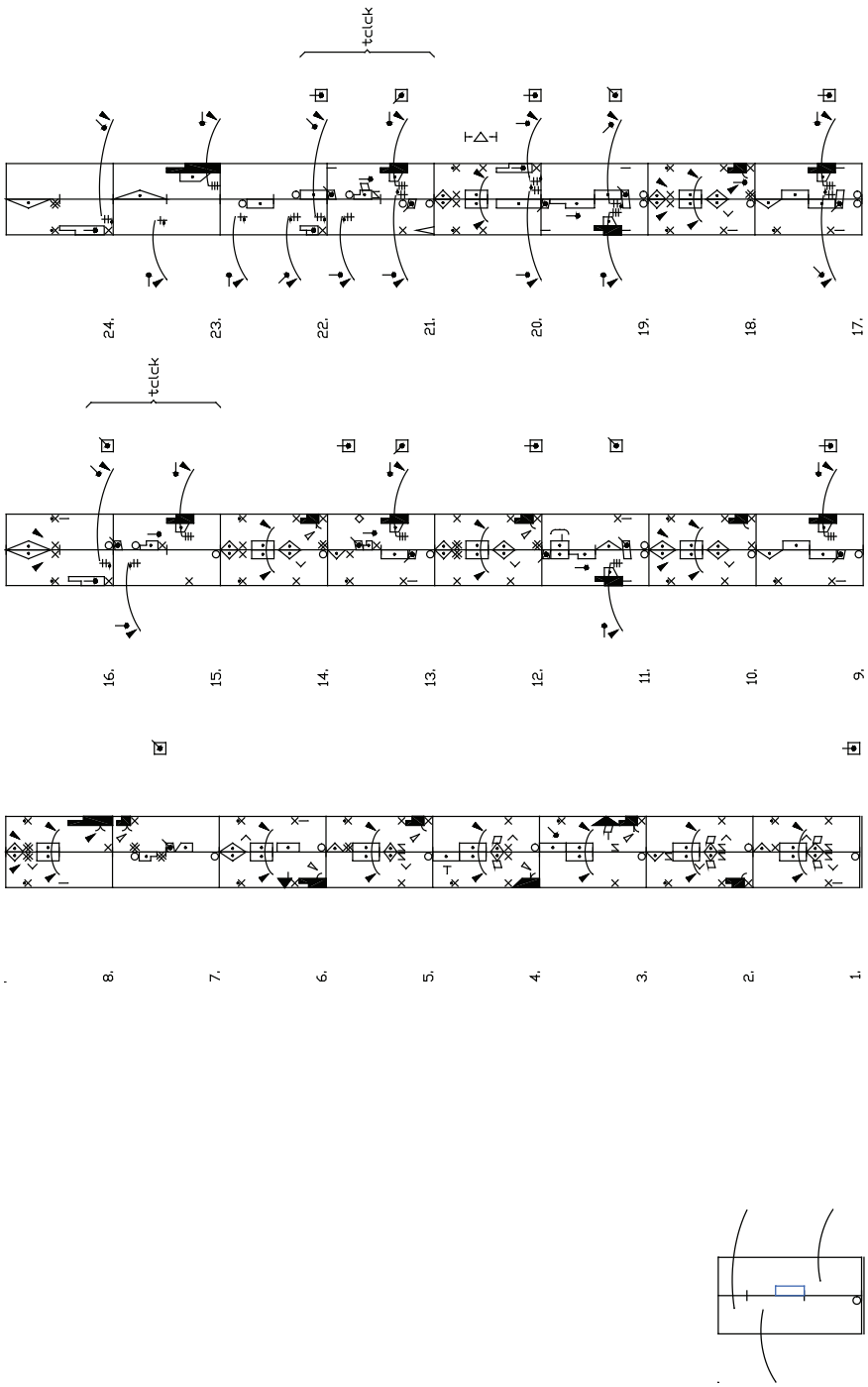


Figure 41. A query and the related hits marked with 'tclck'

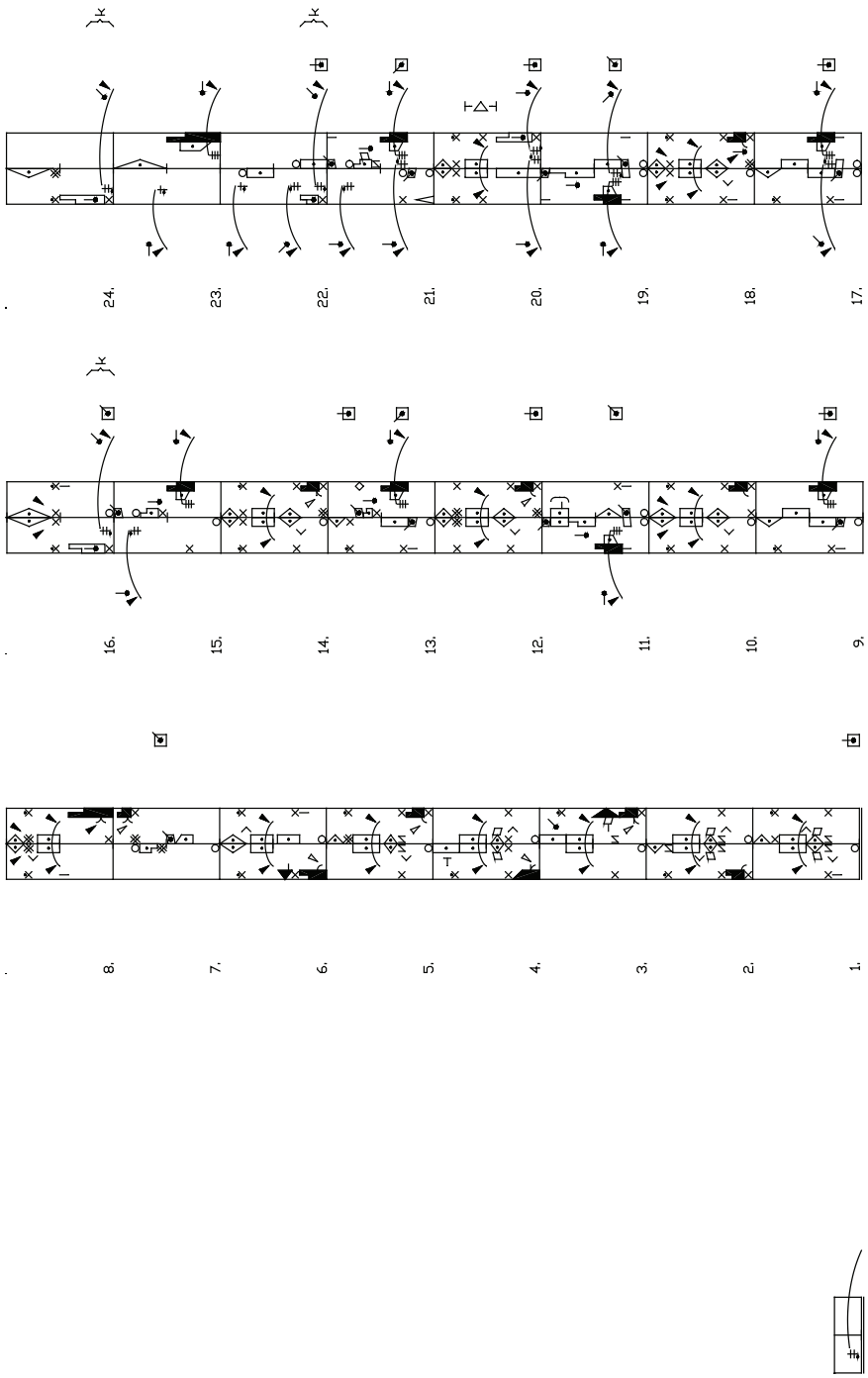


Figure 42. A query and the related hits marked with 'k'

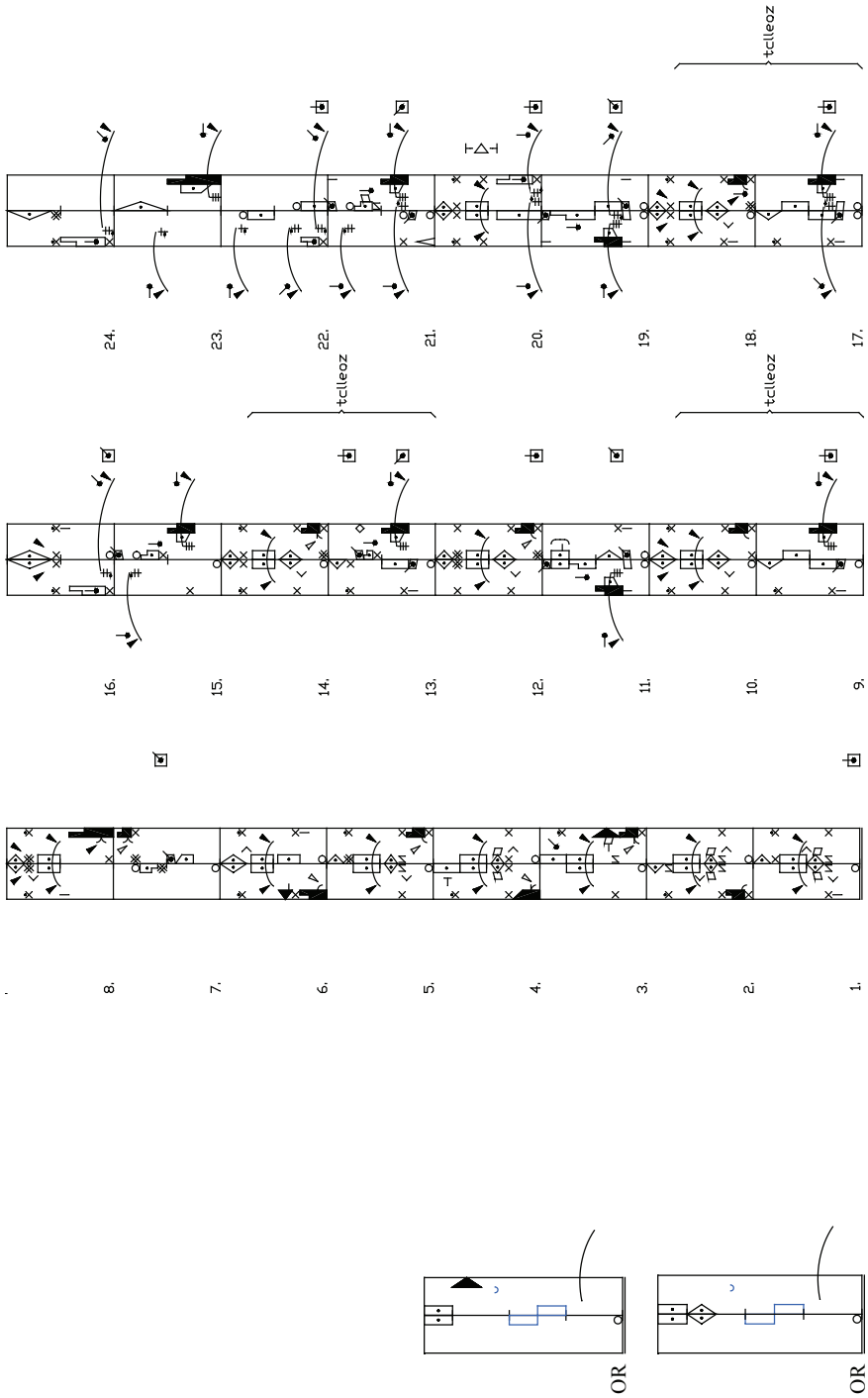


Figure 43. A query and the related hits marked with 'tclleoz'

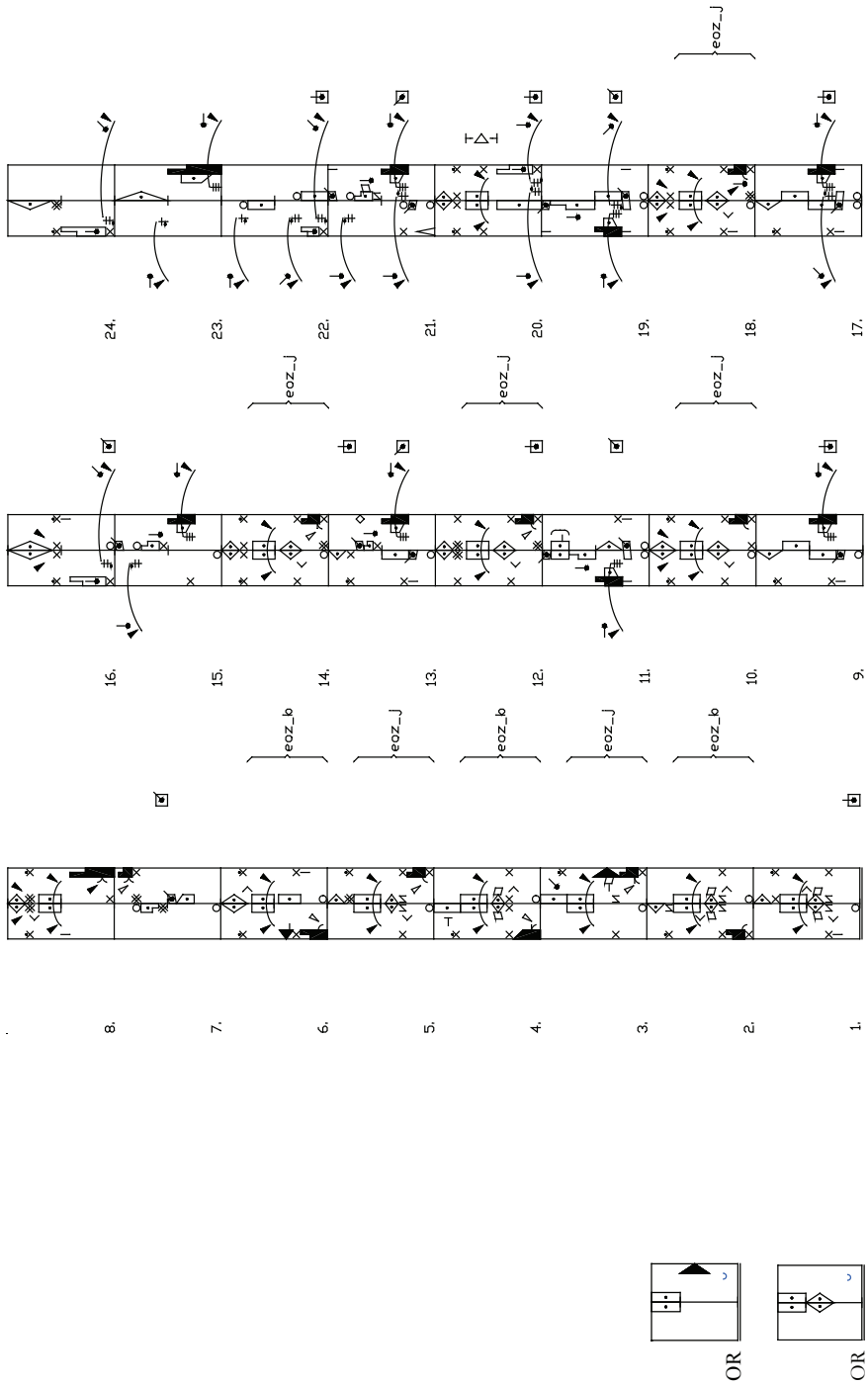


Figure 44. A query and the related hits marked with ‘eoz_j’ and ‘eoz_b’

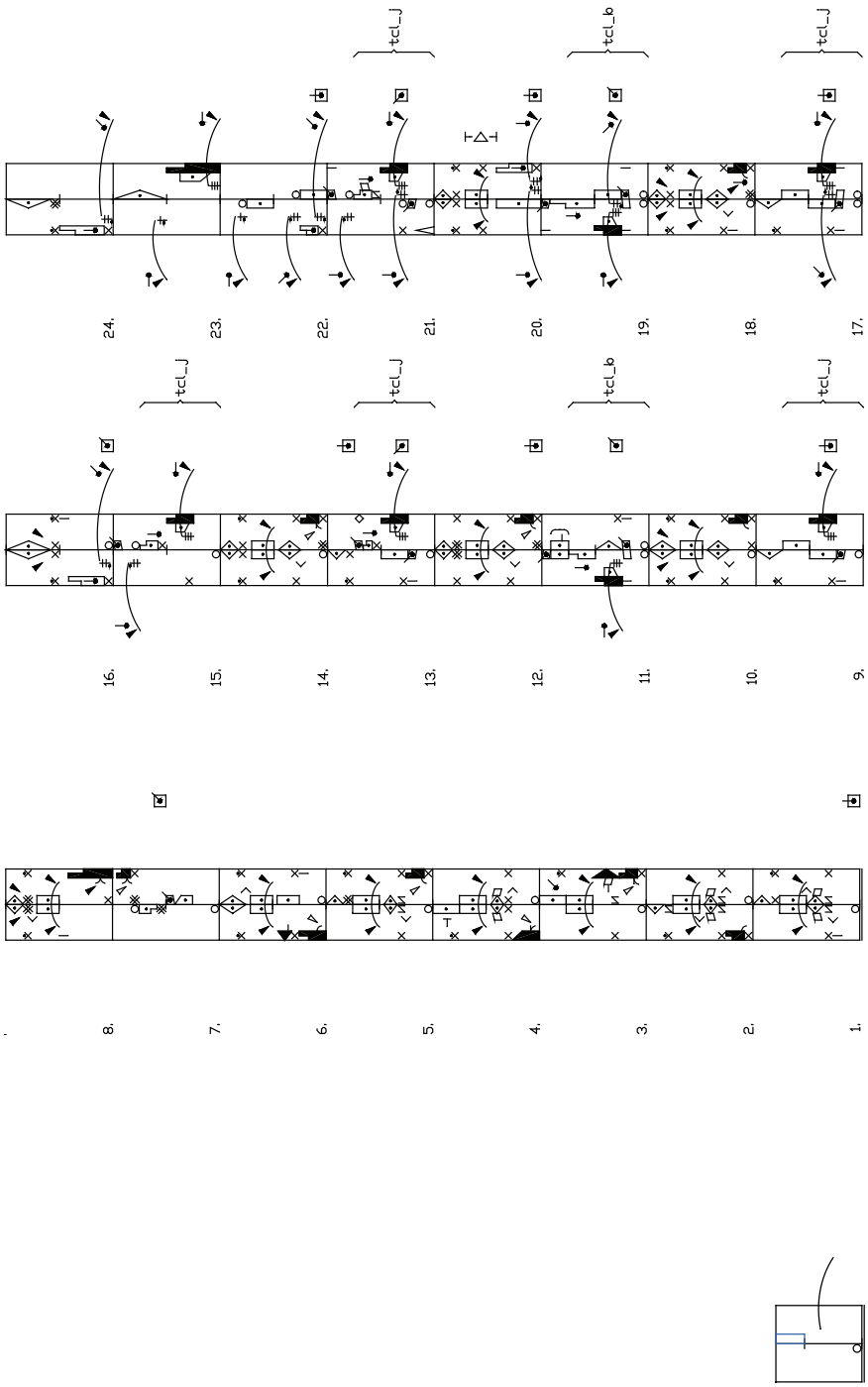


Figure 45. A query and the related hits marked with 'tcl_j' and 'tcl_b'

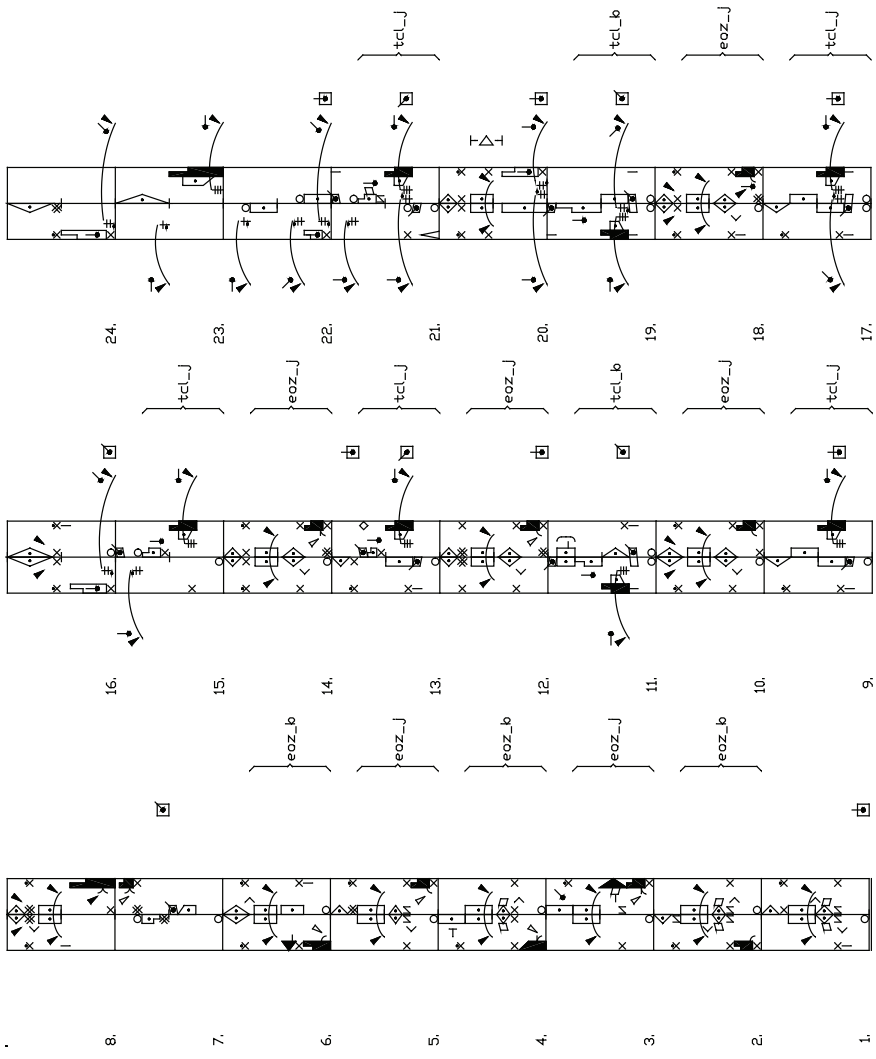


Figure 46. Hits marked with ‘eoz_j’, ‘eoz_b’, ‘tcl_j’ and ‘tcl_b’

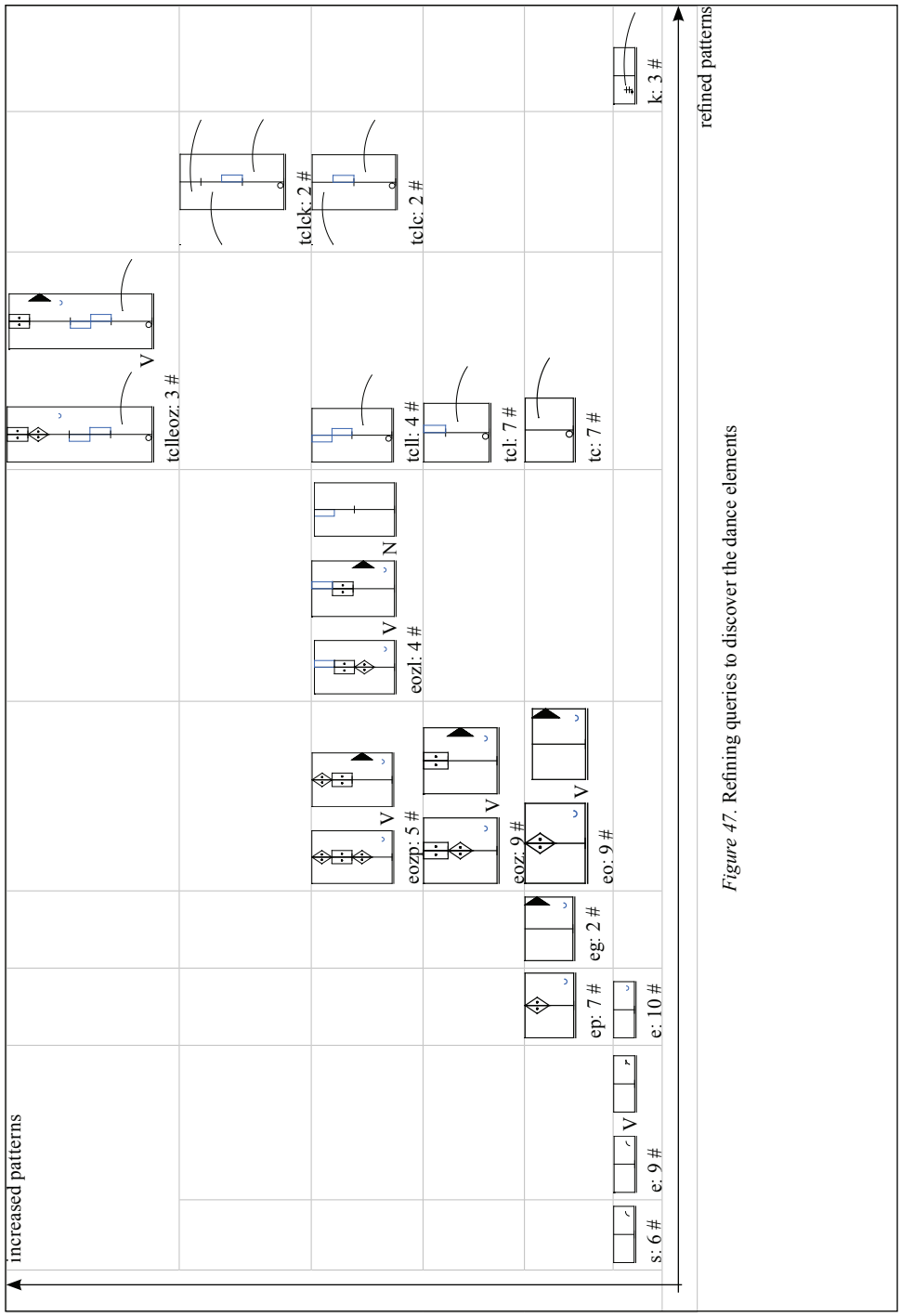


Figure 47. Refining queries to discover the dance elements

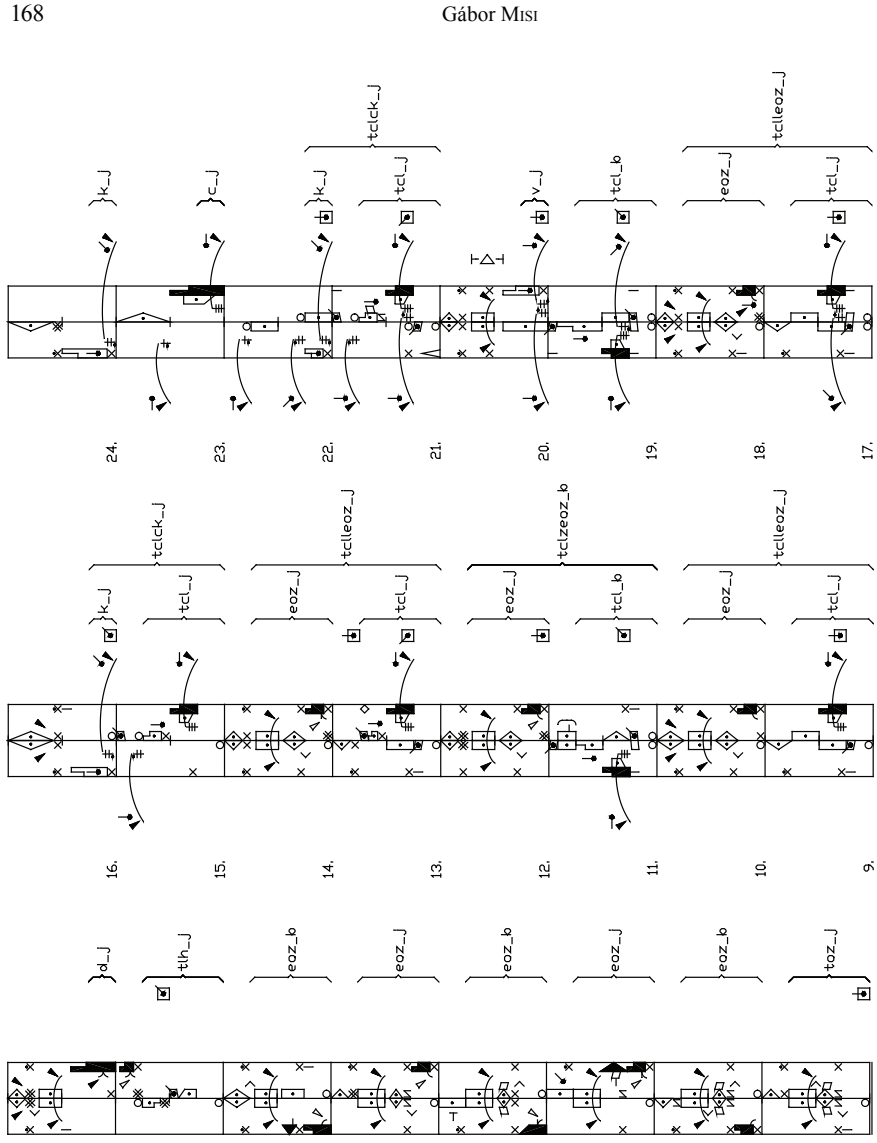


Figure 48. A possible analysis of the simplified corpus

An executable script with Labanatory commands

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; Labanatory AutoCAD script to analyze a short dance corpus ;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
_ BATCH _ MODE 1
_ CLEAR _ MARKERS Yes
_ SET _ MARKER _ PROPERTIES Continuous 18
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
_ COMPOUND _ SEARCH
epz
egz


LJmc
Original Identical Measure OnlyPosition
_ SET _ HIT _ MARKER _ TEXT eoz _ j
_ MARK _ HIT All On
_ COMPOUND _ SEARCH
epz
egz


LJmc
Original Symmetrical Measure OnlyPosition
_ SET _ HIT _ MARKER _ TEXT eoz _ b
_ MARK _ HIT All On
;;;;;;;;;;;;;;;;
_ SIMPLE _ SEARCH
tcl
LJmc
Original Identical Measure OnlyPosition
_ SET _ HIT _ MARKER _ TEXT tcl _ j
_ MARK _ HIT All On
_ SIMPLE _ SEARCH
tcl
LJmc
Original Symmetrical Measure OnlyPosition
_ SET _ HIT _ MARKER _ TEXT tcl _ b
_ MARK _ HIT All On
;;;;;;;;;;;;;;;;
_ SIMPLE _ SEARCH
k
LJmc
```

```
Original Identical Measure OnlyPosition
_ SET _ HIT _ MARKER _ TEXT k _ j
_ MARK _ HIT All On
;;;;;;;;;;;;;;;;
_ SET _ MARKER _ PROPERTIES Continuous 30
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
_ SIMPLE _ SEARCH
tclck
LJmc
Original Identical Measure OnlyPosition
_ SET _ HIT _ MARKER _ TEXT tclck _ j
_ MARK _ HIT All On
;;;;;;;;;;;;;;;;
_ COMPOUND _ SEARCH
tcllepz
tcllegz


LJmc
Original Identical Measure OnlyPosition
_ SET _ HIT _ MARKER _ TEXT tclleoz _ j
_ MARK _ HIT All On
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
_ SET _ MARKER _ PROPERTIES Dashed 18
_ MARK _ INDIVIDUAL LJmc 1 0 3 toz _ j
_ MARK _ INDIVIDUAL LJmc 7 0 3 tlh _ j
_ MARK _ INDIVIDUAL LJmc 8 0 1 d _ j
_ MARK _ INDIVIDUAL LJmc 20 0 1 v _ j
_ MARK _ INDIVIDUAL LJmc 23 0 1 c _ j
_ SET _ MARKER _ PROPERTIES Dashed 30
_ MARK _ INDIVIDUAL LJmc 11 0 7 tclzeoz _ b
_ BATCH _ MODE 0
```