

# Syncopation as Transformation

George Sioros<sup>1</sup> (✉) and Carlos Guedes<sup>2</sup>

<sup>1</sup> Faculdade de Engenharia da Universidade do Porto,  
Rua Doutor Roberto Frias s/n, 4200-465 Porto, Portugal  
gsioros@gmail.com

<sup>2</sup> NYU Abu Dhabi, 5th St., Abu Dhabi, United Arab Emirates  
cag204@nyu.edu

**Abstract.** Syncopation is a rhythmic phenomenon present in various musical styles and cultures. We present here a set of simple rhythmic transformations that can serve as a formalized model for syncopation. The transformations are based on fundamental features of the musical meter and syncopation, as seen from a cognitive and a musical perspective. Based on this model, rhythmic patterns can be organized in tree structures where patterns are interconnected through simple transformations. A Max4Live device is presented as a creative application of the model. It manipulates the syncopation of midi “clips” by automatically de-syncopating and syncopating the midi notes.

**Keywords:** Syncopation · Transformations · Meter · Rhythm · Generation · Analysis

## 1 Introduction

Syncopation is an essential musical phenomenon found in several and diverse styles of western music, as well as in certain non-western music. It is commonly related to rhythmic complexity and music tension [4, 5, 7]. It is well understood among musicians and it has been the subject of research in several fields such as computational musicology and music cognition. Definitions of syncopation often relate syncopation to the musical meter, describing it, for example, as a contradiction to the prevailing meter [18]. Musicological definitions of syncopation focus on its conceptual aspects or describe it as a technique [11, 18]. More formalized definitions approach syncopation as a matter of magnitude [3, 6, 8, 10, 16] and the subject of comparison between rhythmic patterns: patterns are considered to be more syncopated or less syncopated than others. Other studies are limited to specific cases [23] or music styles [8, 24]. However, there has been no formalized model that analyzes and generates syncopation.

Recently we presented an algorithm that allows for the manipulation of syncopation, i.e. a computer algorithm that is able to remove or introduce syncopation in a certain rhythmic pattern in a controlled way [20]. Here, we extend the algorithm to a set of formalized generic transformations that can analyze, generate and manipulate the syncopation in binary patterns. The transformations are derived by the definition of syncopation as a cognitive mechanism related to metrical expectations [7].

We introduce the concept of the syncopation tree as a way of organizing and interconnecting patterns. A syncopation tree is essentially a web of patterns, where

each pattern is connected to other patterns through the application of series of simple transformations. Patterns belonging in the same tree originate from the same root, i.e. they originate from the same non syncopating pattern. The tree structure can be useful in developing models for clustering rhythms together or in defining measures of rhythmic similarity and distance.

The syncopation transformations are accompanied by a method for enhancing the metrical and syncopation feel through dynamic accents. We assign dynamic accents to the onsets on the beat (thus enhancing the metrical feel) and then shift them to off the beat positions (thus enhancing the syncopation feel).

A software application in the form of a Max4Live device is also presented. The software is intended to be used as a tool to explore syncopation both in live performance and in off-line composition. It takes a midi clip and generates different versions of it that range from no syncopation to maximum syncopation. The device can read midi clips and transform them in real time. Transformations are controlled through a simple “syncopation slider”.

In Sect. 2, we provide a brief description of the main syncopation definitions and measures related to our model of syncopation. In Sect. 3, we describe a way of automatically constructing a metrical template and how it must be adapted for modeling syncopation. In Sect. 4, we describe the syncopation transformations for binary patterns and we present the concept a syncopation tree as a structure for organizing rhythmic patterns. In Sect. 5, we extend the transformations to include dynamic accents. In Sect. 6, we present a software application as an example for the creative exploration of syncopation. Finally, in Sect. 7, we present the main conclusions of this study.

## 2 On Measuring Syncopation

Complex music, even when it is not repetitive, it often evokes the sensation of a regular pulse in listeners that becomes evident when they tap in synchrony with the music. Often those pulses feature accents so that some of them are perceived as stronger than the rest. These structured, accented pulses are the basis for a definition of meter as a cognitive mechanism that expresses our expectations about when music events will occur [9]. Syncopation has been described as the feeling of surprise that arises when a rhythmic event that was anticipated at a particular moment did not actually occur [7].

Longuet-Higgins and Lee presented a syncopation model that identifies the syncopation in the pairs of notes and the rests or the tied notes that follow them [14]. Accordingly, a rest or tied note in a strong metrical position preceded by an event in a weaker metrical position constitutes a syncopation. David Huron, in his study of the American popular music [8], used a similar definition using the term “Lacuna” to describe syncopation. Behind both definitions lies the same expectation principle: an event in a weak metrical position is bound to an event in the following strong metrical position and when the expected strong note does not occur the weak one is left hanging [2, p. 295].

Longuet-Higgins and Lee defined syncopation with the aid of metrical weights. The metrical weights correspond directly to the metrical level that each metrical position in

a bar initiates (see Sect. 3 for a more detailed description). The slower the metrical level the higher the metrical weight. In Fig. 1, the metrical levels of a 4/4 m are numbered from 0 to 4. The metrical weights are the negative of those indexes (i.e. from  $-4$  to 0). Syncopations are found when the metrical position of a rest or a tied note has a higher weight than that of the position of the preceding note onset.

W.T. Fitch and A.J. Rosenfeld derived a syncopation measure [3] from the Longuet-Higgins and Lee definition (hereafter LHL). The LHL measure attributes to each syncopation a score that is the difference between the two metrical weights described above. The total syncopation in a rhythmic pattern is the sum of all the syncopation scores. In this paper, when a rhythmic pattern consists of more than one bar, we divide the sum by the number of bars in order for all the results to have comparable scales.

David Temperley explored the uses of syncopation in rock [23], using a definition of syncopation that bears strong resemblance to the Longuet-Higgins and Lee definition and Huron's expectation principle. In his study, syncopation is defined as the displacement of events from metrically strong positions to preceding weaker ones. The syncopation transformations that we describe in this paper are inspired by this study.

As the above definitions of syncopation suggest, we need a formalized way of describing the underlying meter in order to be able to manipulate the syncopation in a rhythmic pattern. We need to construct a metrical template according to the time signature. In the next section we describe such a template, and an automatic way of constructing it. It will serve as the basis for all syncopation manipulations that follow.

### 3 Metrical Template for Syncopation

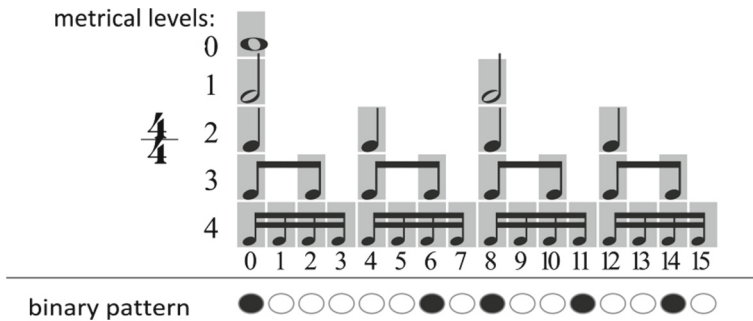
The meter determines a hierarchical structure like the one found in the Generative Theory of Tonal Music by Lerdahl and Jackendoff [12]. Such a structure consists of pulses with different metrical strength values that represent the alternating strong and weak beats commonly found in a musical meter (Fig. 1). Similarly to the metrical structure used by Longuet-Higgins and Lee in their syncopation definition [14], each pulse initiates a metrical level and its metrical strength is proportional to that level. The metrical hierarchy can be thought of as a superposition of layers of pulses with different periods, in which the period of one layer is an integer multiple of all faster ones [17, 25]. Each pulse belongs to the various metrical levels according to the time signature, e.g. the quarter notes or the sixteenth notes found in a 4/4 bar. The pulses constitute a metrical grid which quantizes the time positions of the onsets of the events in a rhythmic pattern resulting into the binary representation of Fig. 1.

The metrical template is constructed automatically for each meter and tempo in which the rhythmic patterns are heard. It is constructed by successively subdividing the bar into faster metrical levels. For example, the 4/4 m can be subdivided first into two half notes, then each half note into two quarter notes, each quarter note into two eighth notes and so on, until the fastest metrical subdivision is reached. That way, the hierarchical character of the template is ensured. The metrical levels are indexed by numbers, referred to as metrical indexes for easy reference, starting with the number 0 for the slower one and increasing as one goes to faster levels. The process of

successively subdividing the bar results in alternating weak and strong pulses, forming a pattern characteristic of each time signature. The stronger a pulse is, the slower the metrical level it belongs to (lower index), so that weak pulses belong to faster metrical levels (higher indexes). In Fig. 1 an example of such a metrical template is given. A detailed description of an automatic way of generating a metrical template for any given time signature can be found in [21].

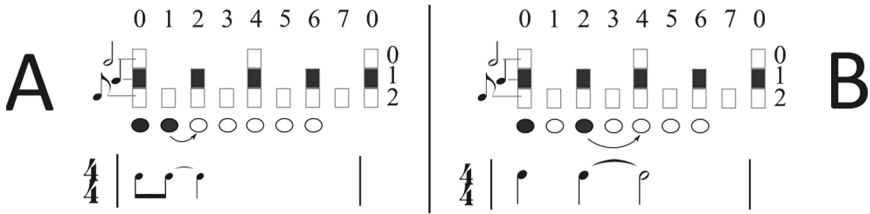
The duration of each metrical subdivision depends on the tempo, e.g. the quarter note at 100 bpm has duration of 600 ms. The lower threshold for the duration of a metrical subdivision has been estimated in several studies to be roughly around 100 ms [13, 17, 19]. The fastest metrical subdivision that we included in the metrical template is the fastest subdivision above that threshold.

An upper threshold for the duration of the metrical levels needs also to be determined. Before determining the upper threshold, we need to define the duration of a syncopation and examine its relation to the metrical levels. The definition of syncopation, as originally formulated by Longuet-Higgins and Lee [14] or Huron [2, p. 295] and also adopted here, attributes syncopation to the pair of an onset on a weak pulse with the following silent strong pulse. One can, therefore, talk about and define the duration of the syncopation as the duration between those two pulses. The duration of the syncopation depends, on one hand, on the metrical levels of the corresponding pulses, and on the other hand, on the tempo. One question that arises is: how does the duration of the syncopation affect the feeling of syncopation?



**Fig. 1.** Top: Example of the construction of a metrical template for a 4/4 m. The meter is successively subdivided generating the 5 metrical levels. The metrical strength (*grey rectangles*) of each metrical position corresponds to the metrical levels it belongs to. *Bottom:* A binary pattern.

The effect of the duration of syncopation on the feeling of syncopation becomes apparent when one tries to de-syncopate a rhythmic pattern. If one tries to de-syncopate the patterns of Fig. 2, he will follow different approaches for pattern A and B. In pattern A, the tied eighth note should clearly be moved to the following quarter note. When it comes to pattern B, the tied quarter note falls on the beat and therefore, intuitively, does not need to be de-syncopated. However, the two cases are identical with respect to the definition of syncopation. The difference between the two is their relation to what is considered to be the beat level (or tactus).

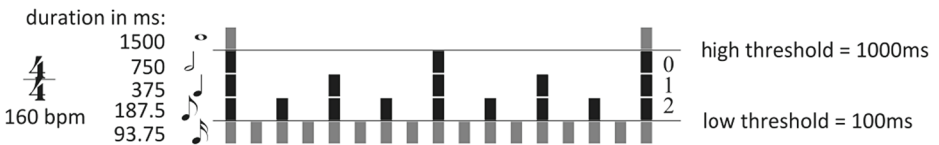


**Fig. 2.** Syncopation at slow metrical levels. A: a pattern syncopating at the eighth note metrical level. B: The same pattern at half speed. Above the two patterns the corresponding metrical template is shown.

During the performance, the listener does not have the score or the notated meter in his mind. He can only tell the difference based on the duration; *A* is performed twice as fast as *B*. For example, if *B* is performed at 100 BPM the duration of the quarter note (q.n.) will be 600 ms. Performing the same pattern twice as fast, i.e. accelerating the q.n. duration to 300 ms, is equivalent to converting the q.n. to eighth notes (*A*). The perceived meter and beat do not change between the two performances. The metrical salience of each level depends predominantly by its period, with a peak salience in the region between 500 ms–1 s [13, 17]. The perceived beat—the most salient metrical level—will be the one with the duration of 600 ms in both performances. The eighth note of pattern *A* falls on the faster metrical level of 300 ms which is significantly less salient. Therefore it is perceived as off the beat and as strongly syncopating.

This aspect of syncopation and its relation to pulse salience has not been thoroughly studied. However, based on our experience, we understand that syncopation involving only slower metrical levels is not felt as strong. In other words, instances of syncopation with duration longer than that of the most salient level seem to be felt significantly less strong. The LHL syncopation measure [14] takes this effect into account indirectly, by giving to the syncopation that involves adjacent metrical levels a relatively small weight. Other syncopation measures, such as the weight-note-to-beat distance (WNBD) [6], relate syncopation directly to the beat level, ignoring all slower metrical levels.

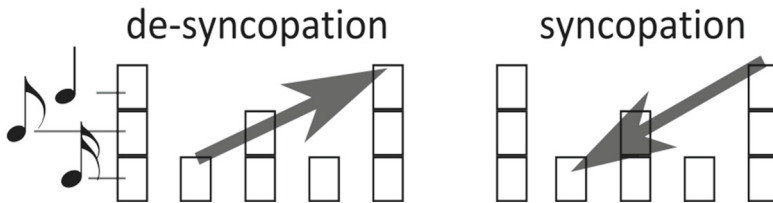
In order to take the above effect of tempo into consideration in the construction of the template, we employ a similar approach to the WNBD by essentially “chopping off” the slower metrical levels. We chose the level that falls in the range between 500 ms and 1 s as the slowest metrical level represented in our structure. For example, in the case of a 4/4 m at 160 bpm (q.n. = 375 ms), the metrical template of Fig. 1 will become as in Fig. 3. At this tempo, only 3 metrical levels survive with corresponding durations of 750 ms (0), 375 ms (1) and 187.5 ms (2).



**Fig. 3.** Example of a metrical template in which very fast metrical levels (below 100 ms duration) and very slow ones (above 1 s duration) are disregarded.

## 4 Shifting Onset Positions

According to the Longuet-Higgins and Lee [14] or Huron’s [7] definition of syncopation, a syncopating event is an event at a weak metrical position that is not followed by an event in the next strong metrical position. If this event was shifted to the strong position the syncopation would be eliminated. With this observation in mind, the event at the weak position can be thought of as belonging to the strong position but been anticipated at an earlier position. Therefore, one can imagine the inverse shift as a way of generating syncopation at a strong metrical position, i.e. by shifting an event found at a strong position to an earlier, weaker pulse (Fig. 4). The transformations described in Sect. 4.1 are such simple shifts of onsets in a binary pattern. The following formalization of the transformations consists of defining the conditions under which such shifts remove the syncopation or generate syncopation in a given binary rhythmic pattern. Applying the formalized transformation on a binary pattern gives rise to the syncopation tree, a network of interconnected patterns, described in Sect. 4.2. In Sect. 4.3 we provide recursive algorithms for automatically generate specific branches. Finally, in Sect. 4.4, we make a preliminary evaluation of the transformations.



**Fig. 4.** The de-syncopation transformation shift events forward to slower metrical levels. The syncopation transformation shifts events backwards to faster metrical levels.

### 4.1 The Transformations

Removing the syncopation is a process of forward shifting of the onsets that syncopate to stronger metrical positions (Fig. 4, left). Each shift of an event results in a new pattern with decreased syncopation. When all events have been shifted to their non-syncopating positions the resulting de-syncopated pattern has no syncopation. We call such a pattern a root. Pulses that belong to slower metrical levels are considered to be stronger, so that the de-syncopation process moves events from fast metrical levels to slower ones.

New syncopation is generated by anticipating the onsets found in strong metrical positions, shifting them to weaker positions (Fig. 4, right). This time, each shift of an event results in a new pattern with increased syncopation. When all “strong” onsets have been moved to weaker positions the resulting syncopated pattern cannot be further syncopated. The syncopation process, opposite to the de-syncopation process, “pushes” events to the faster metrical levels. A detailed description of the syncopation transformation and its inverse—the de-syncopation transformation—follows.

**Syncopation.** The syncopation transformation takes a binary pattern and generates syncopation by anticipating the events found in strong metrical positions. Onsets in pulses that belong to strong metrical positions (slow metrical levels, low metrical indexes) are shifted to preceding pulses belonging to weaker metrical positions (faster metrical levels, higher level indexes).

When syncopating, there might be more than one pulse to shift an onset to. In other words, there might be more than one way to syncopate at a certain metrical position. A strong pulse might be preceded by more than one weak pulses, e.g. a pulse that belongs to the quarter note level is preceded by a pulse at the eighth note level and another one at the sixteenth note level, both belonging in faster metrical levels than the initial one (Fig. 5). We define as the type of syncopation the value of the difference of the metrical levels of the two pulses: the pulse that the onset belongs to originally and the one that it is shifted to<sup>1</sup>. An event is then shifted according to the chosen type to the first pulse that precedes the original position and belongs to the corresponding metrical level. Encoding the syncopation shifts into types of metrical level differences provides the freedom to associate them to metrical positions, to metrical levels or to specific onsets.

For example, if the type is set to 1 (the difference of the metrical indexes = 1), a quarter note is shifted to the preceding eighth note since the eighth note level is one metrical subdivision faster than the quarter note level; while, if the type is set to 2, the same quarter note is shifted to the preceding sixteenth note (Fig. 5). In general, the larger the type value (the difference of the metrical indexes), the shorter it is the duration of the generated syncopation. This is a direct consequence of the alternating character of strong and weak pulses of the metrical template. The number of syncopation types that is available for each pulse depends on how many faster metrical levels exist in the template.

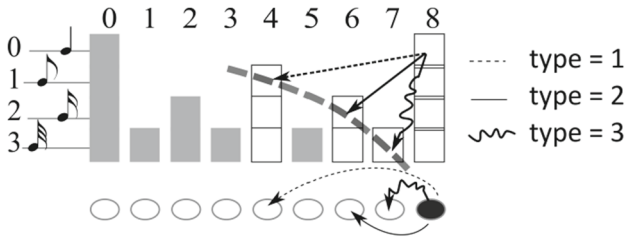
Each syncopation shift is described by a pair of numbers: the pulse that originally carries the onset and the type of syncopation shift that the onset undergoes. They can be thought of as the “coordinates” of the transformation. The pulse index directly corresponds to the “horizontal” coordinate. The type value corresponds to the vertical length of the arrow that represents the shift.

An onset cannot be shifted if one or more onsets block its way. This rule ensures that the order that the events are performed is preserved. In the example of Fig. 5, we cannot apply the syncopation transformation (8, 1) if an onset is found in any of the pulses 5, 6 or 7. The transformation is forbidden.

The above rule might seem at first glance as having no purpose. After all, the onsets in the binary string are indistinguishable from one another and the order in which they are played back is unimportant. Nevertheless, we will later attribute dynamic accents to the onsets. Each one will receive a specific amplitude value and it will carry it in all the generated variations. Thus, onsets will be eventually distinct and therefore their order must be preserved. Moreover, this way the process is more general and can be

---

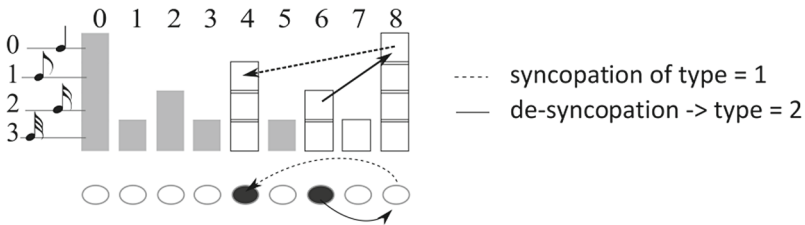
<sup>1</sup> Alternatively we could have encoded the transformation as the pair of pulses, the initial pulse of the onset and the pulse it is shifted to. This way of encoding correctly describes the particular transformation. However, as it will become apparent in the following, using the level differences is a more general and more flexible representation.



**Fig. 5.** An example of the types of syncopation transformations available for an onset at a specific pulse (pulse 8). White square represent the preceding pulses of faster metrical levels that are available for syncopating pulse 8. The rest of the pulses have been greyed out. The corresponding de-syncopation transformation is obtained by reversing the direction of the arrows.

expanded to include other types of phenomenal accents, such as pitch or timbre changes, or other properties of the events that can make them distinct.

**De-syncopation.** The de-syncopation transformation is a direct consequence of the operational definition of syncopation followed in this article [2, p. 295, 4]: syncopation in the binary patterns is found in those events that are placed in a relatively weak metrical position (faster metrical levels, higher metrical indexes) and are not followed by an event in the following stronger position (slow metrical level, low metrical index). In order to de-syncopate such an event, we shift its onset to a position where it no longer syncopates, i.e. to the following strong metrical position that was initially empty, thus evoking the feeling of syncopation.



**Fig. 6.** If the order of onsets was not preserved during the transformations, the 1-1 correspondence of the two transformations would be lost.

The de-syncopation transformation is essentially the inverse of the syncopation transformation. In Fig. 5, onsets that might be found in pulses 4, 6, or 7, can be de-syncopated by reversing the direction of the arrows. As an onset gets de-syncopated, it reveals the type of the syncopation transformation that it had previously undergone. The de-syncopation can be thought of as the analysis process yielding the type of syncopation that was previously generated by a syncopation transformation. An event might need to be shifted more than once in order to be completely de-syncopated. For example, an event found in pulse 5 in Fig. 5 needs to be shifted first to pulse 6 and then to pulse 8 yielding the corresponding syncopation transformations (6, 1) and (8, 1) expressed as pairs of pulses and types.



The above definition of syncopation attributes the syncopation to the event being anticipated. However, the syncopation is only felt at the moment of the following silent pulse and is retrospectively attributed to the event being heard. The following alternative phrasing of the syncopation definition attributes syncopation to the silent pulse where the syncopation is actually felt instead of the onset that initiates it: “syncopation is found in the silent pulses that belong in strong metrical positions (slower levels, lower indexes) and that are preceded by onsets in one of the immediately preceding weaker metrical positions”. Thus, de-syncopation must be applied to the silent pulse by shifting the preceding onset. In Fig. 6, de-syncopating pulse 8 is done by shifting the onset of pulse 6 to pulse 8 (solid arrow), yielding the corresponding syncopation type and completing the syncopation pair (8, 2).

The de-syncopation transformations reveal a deeper reason for preserving the order of onsets when syncopating and forbidding certain transformations. If one could jump over an existing onset when syncopating, then the transformations could not have a 1-1 correspondence. Imagine an onset originally existing in pulse 8 of Fig. 6 and apply the (8, 1) syncopation transformation (dashed arrow). The de-syncopation of pulse 8 would result in a different pattern than the one we started with. Moreover, it would return a different syncopation transformation—of type 2, as we already saw. Thus, the two transformations would not be reversible if two onsets could reverse their order.

The described transformation guarantees that the resulting pattern will not syncopate. This is not true for other similar transformations. In some cases, one could de-syncopate off-beat events by shifting them to the preceding on-beat position instead of the following one. However, the two directions are not equivalent. The operational definition of syncopation attributes the syncopation feel to the lack of an event in the following strong position. The forward shifting of events always guarantees the resulting pattern will not syncopate, as the “empty” strong pulse that was causing previously the syncopation will now receive an onset.

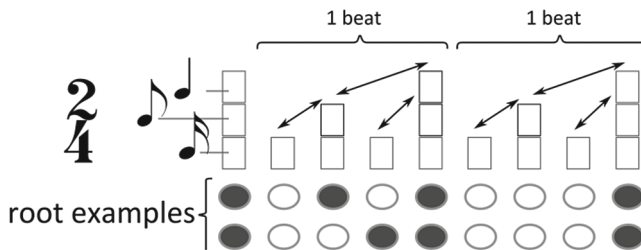


Fig. 7. The syncopation shifts can only move onsets in the duration of a single beat.

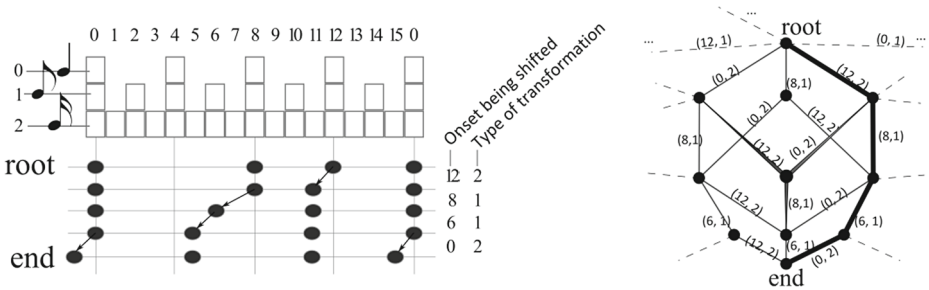
An important property of the transformations comes about the structure of the metrical template of alternating strong and weak pulses and the prominent role given to the beat—the most salient of the metrical levels. As we have seen, the transformations can shift events either forward to slower metrical levels or backwards to faster metrical levels. Since the beat level is the slowest metrical level included in the template, each transformation has local character; it displaces events within the duration of a single

beat<sup>2</sup> (see Fig. 7). Thus, any pattern longer than a beat can be considered a concatenation of shorter and independent—with regard to the syncopation transformations—single beat patterns.

### 4.2 The Syncopation Tree

The two transformations described above can be used to generate paths of syncopation transformations, i.e. a sequence of specific transformation steps, each of which represents the displacement of a single event. To be consistent with our nomenclature, we call such a path a “branch”; for example, the branch that connects a syncopated pattern to its non-syncopating root or to some other syncopated pattern. The branches that connect all possible patterns that originate from the same root form a syncopation tree. In this section we are providing with an overview of the generation of syncopation trees and some of their properties.

On the left side of Fig. 8, we present an example of how a branch is generated starting from the root pattern and applying a series of transformations until we reach the end of the branch where no onset can be syncopated further. However, one could start from any pattern and apply syncopation and de-syncopation transformations to reach another pattern of the same tree. The root pattern of the example has 3 events and is considered to be a repeating loop (the first event is repeated at the end). The branch consists of a set of transformations of specific types indicated by the two numbers next to each pattern. The specific branch is generated by applying the transformations in a particular order. The pairs of numbers form two arrays: (i) one contains the indexes of the pulses in the particular order of the transformations and (ii) the second contains the



**Fig. 8.** Left: An example of a branch starting from the root pattern and finishing to an end pattern that cannot be further syncopated. Right: Example of a syncopation tree (partially shown). The patterns are shown as dots and the lines connecting them correspond to the indicated transformations. The thick line corresponds to the branch shown on the left. The patterns shown are connected with the same set of syncopation transformations applied in a different order.

<sup>2</sup> The duration of the beat is offset by a single pulse to what commonly is considered the duration of the beat (Fig. 7). The beat duration here ends ON the beat and includes all preceding pulses between the current and the previous beat.

type of each transformation. The two arrays completely describe the branch. We refer to such coupled arrays as the branch arrays.

The order of the transformations can change to generate a different branch. As long as the type of each syncopation is kept the same, the end pattern will always be the same. In other words, as long as we include the same transformations but apply them in a different order, we will reach the same end pattern<sup>3</sup>. However, it must be noted that not all permutations are possible. Here, the transformation (8, 1) must always be applied before the (6, 1) otherwise there will be no onset in pulse 6 to be shifted. Some intermediate patterns will also be shared among the branches. In the right side of Fig. 8, we show the branches of the corresponding syncopation tree that include all possible permutations of the transformations in the left (solid lines, the thick solid line is the branch shown in detail at the left).

Each dot in the right part of the figure represents a specific pattern and is connected to the root pattern with a specific number of steps, independently of the exact branch that one might follow. This is a general property of all syncopation trees and not of this particular example. The root pattern as well as the number of de-syncopation steps for a given syncopated pattern is independent of the order in which the pulses are de-syncopated. Only the intermediate patterns differ.

The entire tree is formed by several branches like the ones shown in Fig. 8. These branches form a network that begins at the root and has several ends. The number of ends depends on the number of possible transformations that can be performed within each beat duration. In our example, the 3 onsets belong to different beats and can undergo 2 different transformations each: either the  $(i, 1)$  or the  $(i, 2)$  followed by the  $(i - 2, 1)$ , where  $i$  is the index of the pulse. Therefore, we have  $2^3 = 8$  different ends in this tree. In Fig. 8, we show only the branches that lead to the end pattern shown on the left. The branches that lead to the rest of the ends are connected to the patterns of the figure through the dashed lines.

The complexity of the tree depends on the root pattern and the metrical template used. Visualization of the tree can become challenging as the number of the connections of each pattern can be large. One can imagine the tree as a three dimensional structure with the root pattern in the center of a sphere and the end patterns on its surface. The intermediate patterns form a web in the interior of the sphere.

Any two patterns in the tree have at least one branch that connects them, passing through several other patterns. Generating such branches is equivalent to analyzing the relations of the generated patterns.

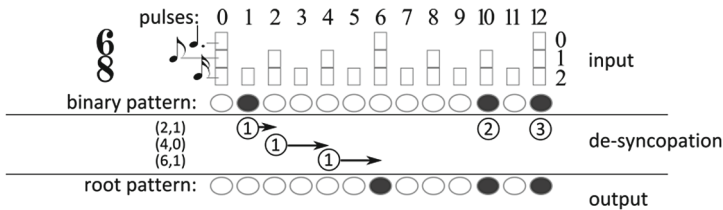
Since events cannot be shifted from one beat to another, each beat must contain the same number of events in all patterns of the same tree. As a consequence, patterns with different distribution of events in the beats will have different roots and therefore belong in different trees. However, more than one root patterns exist that have the same number of events in each of their beats (Fig. 7). As consequence<sup>3</sup>, the distribution of events in beats is not the only criterion for two patterns to belong in the same tree.

---

<sup>3</sup> In order for the end pattern to be the same, all transformations in a given permutation must result in the shift of an event, i.e. the permutation should not result in forbidden or blocked transformations.

### 4.3 Recursive Algorithms

**De-syncopation.** The generation of a branch consists in determining the corresponding series of syncopation transformations expressed as a branch array, i.e. an order array coupled to a type array. Such arrays can be generated by a de-syncopation process, i.e. by recursively applying a de-syncopation transformation for each pulse that is found to syncopate until there is no syncopating pulse. Using the de-syncopation output for a given rhythmic pattern, one can generate all intermediate patterns from the corresponding root pattern to the original input pattern. As we already saw, the root pattern is independent of the order in which the pattern is de-syncopated.



**Fig. 9.** Illustration of a recursive de-syncopation process. The metrical template shown corresponds to a 6/8 m at 180 bpm (q.n.d. = 1000 ms). Each onset of the binary pattern is numbered in a circle. The de-syncopated process is shown as arrows.

Figure 9 illustrates the process through an example. Pulses are examined in a certain order and when a silent pulse is preceded by an onset in a faster metrical level, the onset is shifted as described in Sect. 4.1. In this example, the pattern is scanned from right to left. However, the order in which the pulses are examined can be, in principle, freely chosen. The first pulse examined is pulse 12, which carries an onset and therefore does not syncopate. Pulse 11 as well as all pulses of the fastest metrical level cannot syncopate. Pulse 10 is also ignored, as it carries an onset. Pulse 8, 6 and 4 do not syncopate since they are not preceded by an onset in a faster metrical level. The first silent pulse found to syncopate is pulse 2 that belongs to the eight note level. Event ① precedes it in pulse 1 (sixteenth note level). Therefore it is shifted to pulse 2 yielding the undergone syncopation transformation (2, 1).

The process restarts from pulse 12 and now the first syncopating pulse is 4. This is a special case of syncopation where the preceding event is in a pulse of the same metrical level as the pulse in question. However, they both belong in a subdivision of the beat, i.e. they do not belong in the slowest metrical level included in the template. In that case, the event needs to be shifted and the type of syncopation is 0 yielding the (4, 0) transformation. This special case can occur in meters with a ternary subdivision of the beat such as the 6/8. In the final round, pulse 6 needs to be de-syncopated resulting in the (6, 1) transformation.

The following pseudo code illustrates the basic steps in the de-syncopation algorithm:

```

##inputs
bool PATTERN = {O1, O2, ..., On}
int LEVELS = {L1, L2, ..., Ln}
int ORDER = {P1, P2, ..., Pm}
##outputs
int outORDER = {empty array}
int outTYPE = {empty array}
## DE-SYNCOPAION LOOP
Repeat
  For each position p in ORDER
    if (PATTERN[p]==FALSE AND LEVELS[p]<=max(LEVELS))
      r = Find_Preceding_Pulse(p)
      if (PATTERN [r to p-1 r]==FALSE)
        PATTERN[r]=FALSE;
        PATTERN[p]=TRUE//Shift onset r->p
        Output PATTERN
        Append p in outORDER
        Append (LEVELS[p]- LEVELS[r])in outTYPE
      end
  end
Until no onsets can be shifted
Output outORDER, outTYPE

```

The above code makes use of a subroutine for finding the preceding onset, if any:

```

Find_Preceding_Pulse(p)
LMIN = LEVELS[p]
p--; CurrentML = LEVELS[p]
While (LEVELS[p] > LMIN AND PATTERN[i]==FALSE)
  p--; CurrentML = min(LEVELS[p], CurrentML)
End
If (PATTERN[p]==TRUE AND LEVELS[p]==CurrentML AND
    L[p]>LMIN)
  Return p
Else Return -1
End

```

The above code receives as input three arrays: (i) the pattern as a binary string (PATTERN), (ii) the metrical template as an array of the metrical indexes (LEVELS) and (iii) the ORDER array which represents the order in which the pulses should be scanned and the found syncopations should be eliminated. The algorithm outputs the new binary pattern with each shift of an event. At the same time it stores the details of each shift in two arrays. The outORDER array contains the pulse of the shift and the

outTYPE array contains the metrical level difference. The two arrays together comprise the branch arrays generated by the de-syncopation and describe the transformations applied to the input pattern. They are output at the end of the de-syncopation loop. In the example of Fig. 9 the two arrays would be:

```
outORDER = 2, 4, 6
outTYPE  = 1, 0, 1
```

A couple of examples of default values for the input order array would be the pulses in their natural order, from left to right (0, 1, 2, etc.), or in their order of importance, e.g. according to their metrical level, de-syncopating first the faster metrical levels, or according to the indispensability values of Clarence Barlow [1, 2]. Not all pulses need to be contained in the order array, since not all pulses carry onsets that need to be de-syncopated. However, it is important to ensure that all pulses that will syncopate in any of the intermediate steps of the de-syncopation process will be included in the input order array. As long as this is ensured, the order of transformations only affects the intermediate steps. The root pattern and the instances of syncopation found are the same for any order. The number of elements in the output branch arrays, as well as the values of each pair in the arrays, is unaffected by the input order. Only their position in the arrays could in certain cases change, depending on the pattern. In the example of Fig. 9, the result is always the same; all output arrays and patterns will be exactly the same, independently of the input order.

**Syncopation.** In order to automatically syncopate an input pattern, one can apply repeatedly a series of syncopation transformations until no onsets can be further shifted. The branch arrays for the syncopation process need to include all pulses that carry onsets in any of the output patterns of the applied transformations. That will ensure that when the transformations are applied recursively all onsets will be shifted in syncopating positions and the end of the branch will be reached.

The process is similar to the one shown on the left side of Fig. 8. However, in that example, the transformations were applied once and were particular to the generated branch. In a recursive syncopation process, the branch arrays can be generic and can automatically generate a complete branch from root to end pattern. For example, a default order of transformations can be generated by starting from the pulses in the slowest metrical level and continue to the faster ones until all pulses are included (the fastest metrical level can be ignored). After applying the possible transformations to all the pulses in the array, the process repeats until no onset can be displaced.

For each pulse, a type of transformation should also be assigned. Several options for default values are available. The simplest is a constant value for all pulses. The types can also be chosen according the metrical level of each pulse. For example, all notes could be shifted to the preceding sixteenth note. That would be a type 2 for the quarter notes and a type 1 for the eighth notes.

In cases where the level difference expressed in the type of syncopation results in a metrical level faster than the fastest metrical subdivision included in the metrical template, then this fastest metrical subdivision is used disregarding the type. For example, if the type is set to 2 for an onset at the eighth note level, it should be shifted to the preceding thirty-second note. If the metrical template only goes as fast as the

sixteenth note level, then the onset should be shifted to the preceding sixteenth note, disregarding the type value found in the array.

If one wishes to reverse the transformations generated by the de-syncopation algorithm, he only needs to reverse the two branch arrays, outORDER and outTYPE, produced by the recursive de-syncopation process. Performing the transformation once on the root pattern generates the exact steps of the de-syncopation.

Moreover, one can use the arrays produced during the de-syncopation of a pattern as a point of departure for creatively generating transformations for another pattern. Transformations can even be mapped across different meters, by mapping the order and types to the metrical levels instead of the pulses. For example, during a de-syncopation process, the events found at the eight note metrical level were shifted to the quarter note level (type 1). They were also shifted before any other events at faster levels. Those transformations can be mapped and used when syncopating another pattern with a different metrical template. Pulses at the same metrical level could be put last in the order array and receive the same value of type transformation, i.e. shifts to the preceding pulses of the next faster subdivision. The values for the type array can also be “learned”, e.g. by analyzing a collection of rhythms instead of a single pattern and then generating appropriate order and type values. Of course such mappings are not always meaningful.

The following pseudo code illustrates the basic steps in the syncopation algorithm:

```

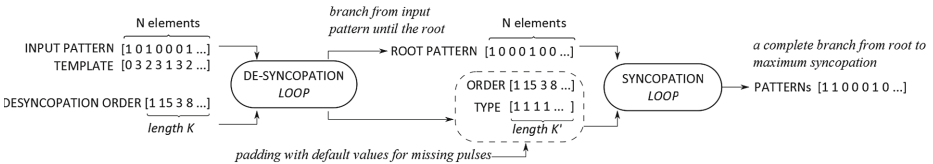
##inputs
bool PATTERN = {O1, O2, ..., On}
int LEVELS = {L1, L2, ..., Ln}
int ORDER = {P1, P2, ..., Pk}
int TYPE = {T1, T2, ..., Tk}
int MAXrepeat = maximum number of repetitions
int step = 0;
##outputs
int outORDER = {empty array}
int outTYPE = {empty array}
## SYNCOPATION LOOP
Repeat
  For each position p[i] in ORDER //at index i of ORDER
    if PATTERN[p[i]]==TRUE AND
      LEVELS[p[i]]!=maximum[LEVELS]
      Find preceding position r:
        (LEVELS[r] = LEVELS[p[i]] + TYPE[i])
      if PATTERN[p[i]-1 to r]==FALSE
        PATTERN[p[i]]=FALSE;
        PATTERN[r]=TRUE//Shift p[i]-> r
        Output PATTERN
        Append p[i]in outORDER
        Append (LEVELS[r]- LEVELS[p[i]])in outTYPE
    end
  step += 1
Until no onsets can be shifted OR step >= MAXrepeat
Output outORDER, outTYPE

```

A “*find preceding position*” subroutine is used to find the preceding pulse that belongs to the faster metrical level. The subroutine is similar to the one used in the de-syncopation algorithm with the difference that it is looking for a silent pulse and that the pulse belongs to specific metrical level determined by the type of transformation.

At the end of the algorithm shown above, a pair of branch arrays results as output. These arrays may differ from the input ones. As the pattern is scanned according to the input order, some transformations might be blocked or skipped when an onset is not found resulting in a different final order. The two arrays, outORDER and outTYPE, contain the actual transformations that were performed.

Figure 10 presents a schematic overview of the entire process of generating a complete branch that passes through a particular pattern. The process is divided in an analysis and a generation stage. First, the de-syncopation of the input pattern to its root provides a detailed analysis of the syncopation of the given pattern. Second, a complete branch is generated, starting with the root pattern, reaching the input pattern and expanding to the other end, to a maximally syncopated pattern. The first part is performed by the de-syncopation algorithm and the second by the syncopation algorithm.



**Fig. 10.** Overview of the generation of a branch of syncopation transformations that passes through a given input rhythmic pattern

It is important to note that the de-syncopation process functions as an analysis stage, even though it actually generates all the intermediates between the root and the input pattern as part of the analysis. The root pattern together with the branch arrays of order and types gives a complete picture of the generated patterns and their relations in a compressed form. The mere collection of the actual intermediate patterns can be thought of as a “by-product” of the process.

The second part of the process generates the entire branch beginning with the root. By altering the order of elements in the syncopation arrays, one can generate a different branch that might not pass by the given input pattern but it will end at the same end pattern. The patterns found in that branch will share the same types of syncopations, although the patterns in their entirety would not be identical. However, altering the type values leads to a completely different branch.

#### 4.4 Preliminary Evaluation of Transformations

In this section we evaluate the syncopation transformations described in Sects. 4.1 and 4.3 in practice, i.e. with actual musical material and not just theoretical binary patterns. We applied the transformations on a collection of MIDI drum loops. Then, we used the LHL syncopation metric described in Sect. 3 to measure the syncopation in the original

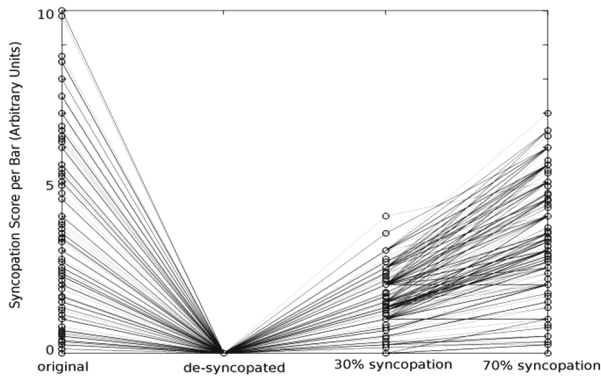


loops and in three more versions that have undergone the syncopation transformations: (1) the root pattern, (2) a syncopated version with 30 % of the total transformations applied, and (3) a syncopated version with 70 % of the total transformations applied. The two syncopated versions were automatically syncopated with a default branch array. Therefore, they belong to different branches than the corresponding original patterns.

The MIDI collection was taken from [15] and consists of 160 drum loops of different music styles. The MIDI files were segregated into three separate MIDI streams according to the MIDI note numbers that correspond to the kick drums, snare drums and hi-hats. All 480 loops ( $3 \times 160$ ) were in a 4/4 m and quantized to the 16th note grid and they were converted to binary patterns before applying any transformation. In all cases we used a metrical template that corresponds to a 4/4 m at 100 bpm.

First, they were de-syncopated according to the process described in Sect. 4.3. Second, the resulted de-syncopated patterns were re-syncopated with 30 % and 70 % of total of the possible syncopations applied. The order array was generated by shuffling the indexes of all pulses. The type of the syncopation process was set to 2, forcing the syncopation shifts always to two metrical subdivisions faster. We used the LHL metric to measure the syncopation in all versions of each of the 480 loops. An overview of the results is shown in Fig. 11.

The de-syncopation and re-syncopation algorithms performed as expected. The de-syncopation algorithm removed completely the syncopation in all 480 binary patterns (syncopation score = 0). This was expected as the de-syncopation process directly corresponds to the way onsets are matched to following silent pulses in the definition of syncopation in the LHL algorithm.



**Fig. 11.** Syncopation measurements using the LHL algorithm on 4 different versions of the patterns found in the 480 binary patterns. Each circle represents a single measurement. The grey lines connect the measurements for the 4 different versions of each pattern. As the number of measurements is large, several lines naturally overlap. The darker the line is, the more the overlaps.

The re-syncopated process increased gradually the syncopation. The 30 % transformation increased the syncopation in the majority of the patterns (in 469 out of 480 patterns). The 70 % transformation increased further the syncopation for the majority of the patterns (in 445 out of 480 patterns). No pattern had less syncopation in the 70 % than in the 30 % transformations. In a few exceptions, the patterns had the same syncopation score for the de-syncopated and 30 % transformation (11 patterns), or for the 30 % and 70 % transformations (35 patterns) (horizontal lines in Fig. 11). In 6 patterns out of the total 480 the re-syncopation transformations did not create any syncopation.

The exact relation between the syncopation scores and the percentage of the applied transformations depends on two factors: (1) the number of onsets per bar in the pattern that can actually be shifted and are not blocked by other onsets and (2) their metrical positions. When only a small number of events can be shifted there are accordingly few steps between the de-syncopated version and the fully syncopated version (100 % of the transformations applied). The number of events that can be shifted in the pattern strongly depends on the density of events per bar. For very low density (e.g. only a couple of events per bar) or very high density (when almost all metrical positions are occupied) the onsets available for syncopation are very few. The evaluation shows that the syncopation transformations increase the syncopation with each step, even when the available steps are very few.

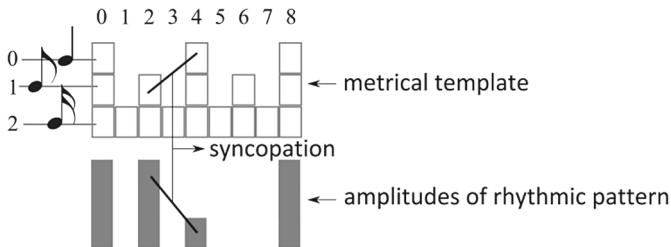


Fig. 12. An example of a temporary mismatch of the dynamic and metrical accents.

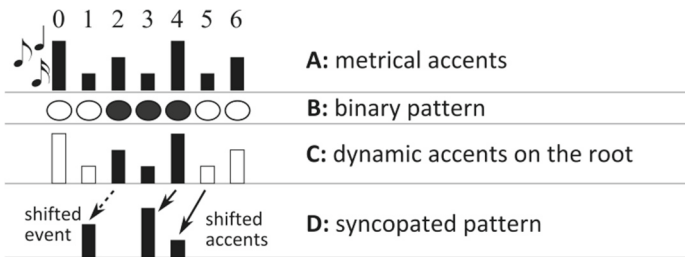
## 5 Transforming Dynamic Accents

Dynamic accents are produced by relative changes in the loudness of serial events. When an event is louder than its neighbor is felt as stressed. One can extend the definition of syncopation for binary patterns to account for dynamic accents. When an event at a weak metrical position is accented relatively to an event in the following strong position, syncopation is felt. The syncopation will be felt stronger when the onset in the weak position is more stressed and when the metrical levels of the two pulses are further apart. The syncopation measure proposed by Sioros and Guedes [22] uses this principle to measure syncopation in accented patterns.

Loudness changes are distinguished from metrical accents [12]. They are part of larger group of physical accents that arise from changes in some dimension of the sounding events. They are often referred to as phenomenal accents. With this

distinction in mind, syncopation can be described as “a temporary mismatch of the phenomenal and metrical accents”.

In Fig. 12, an example of a syncopation arising from dynamic accents is shown. A loud event in pulse 0 is followed by an equally loud event in pulse 2. Although pulse 2 belongs in a weak metrical position, it is not against our metrical expectations to hear an onset there. However, the loud event in pulse 2 will raise our expectations for a loud event in pulse 4. When a much quieter event actually follows, the previous event sounds relatively stressed. This stress is against our metrical expectations and the feeling of syncopation is evoked. As the difference in loudness gets greater the feeling is more intense. In the extreme case that no onset is heard on the strong pulse, our definition of syncopation is equivalent to the definitions for binary patterns (see Sect. 2).



**Fig. 13.** Example illustrating the process of assigning and manipulating dynamic accents. **A:** Metrical accents calculated based on the metrical template. **B:** A binary, de-syncopated pattern. **C:** Dynamic accents on the events in their de-syncopated positions. **D:** Syncopated pattern. The event on pulse 2 is now shifted to pulse 1 (*dashed arrow*). The events on pulses 3 and 4 cannot be shifted. Instead, the accents on pulse 3 and 4 are shifted (*solid arrows*).

We introduce dynamic accents to the binary patterns for two reasons: (i) to enhance the metrical feel by accenting the onsets on strong pulses and (ii) to generate syncopation on onsets that could not be shifted by a syncopation transformation such as the ones described in Sect. 4.

Each onset is assigned a specific dynamic accent in the range between 0 (a silent onset) and 1 (loudest possible) relative to the metrical strength of its position. In that way, a binary pattern becomes a sequence of amplitude values. In Sect. 5.1, the accents are assigned to the onsets according to their de-syncopated positions enhancing that way the metrical feel (phenomenal accents matching the metrical accents). In Sect. 5.2, the accents are “shifted” to onsets in weaker metrical position generating syncopation (mismatch of phenomenal and metrical accents).

## 5.1 Generating Accents

The metrical feel is enhanced when phenomenal accents coincide with the metrical accents, i.e. when changes in the loudness, pitch etc. coincide with the alternation of strong and weak pulses characteristic of the meter. Based on this principle, each onset

is assigned an amplitude value relative to the metrical strength of the pulse that the onset belongs to.

The amplitudes are assigned to the root pattern and the onsets carry the accents to all syncopating positions. This is because a syncopating onset can be found in weak metrical positions. If we were to assign the amplitude according to this weak position, we would weaken the syncopation without necessarily enhancing the metrical feel since the onset will still be syncopating. Furthermore, if the onset is shifted to its de-syncopated position carrying a weak accent, the syncopation might not be completely eliminated.

The amplitudes are assigned to the onsets themselves and not to the pulses, so that when the onsets are shifted to different positions in the various syncopated versions, they still carry their accents from the root pattern (Fig. 13). For example, an onset found at the beat or quarter note level in the root pattern will still be a loud onset with a high amplitude value even when it is shifted to a faster metrical level. In that way, the feeling of syncopation is intensified. The syncopating event is distinct from other events that are found in fast metrical levels but that do not syncopate. Those non-syncopating events receive lower amplitudes since they belong in fast metrical levels in the root pattern.

The following mathematical formula is used to calculate the amplitudes of onsets:

$$A(i) = C^{L(p_i)} \quad (1)$$

where  $A(i)$  (range: 0–1) is the amplitude of event  $i$ ,  $C$  is a parameter that controls the “contrast” between strong and weak metrical positions,  $p_i$  is the pulse that event  $i$  belongs to (in the root pattern) and  $L(p_i)$  is the metrical level of that position. The parameter  $C$  ranges between 0 (only the slowest metrical level survives) and 1 (all metrical levels receive the same amplitude).

## 5.2 Generating Syncopation

Dynamic accents can generate syncopation when the actual displacement of onsets is not possible. Imagine the extreme case where all pulses, even at the fastest metrical subdivision, carry an onset. That would be, for example, the case of drum rolls. In such cases, onsets cannot be shifted since there are no empty pulses available and the only way to reveal the meter or contradict it is through the use of accents. In order to introduce syncopation to such a pattern, we stress certain off-beat onsets while attenuating the ones that follow them on the beat. For those onsets, the phenomenal accents would not coincide with the metrical accents, and that mismatch of phenomenal to metrical accents would be felt as syncopation.

The accents of the onsets in their de-syncopated positions are proportional to the metrical strength of the positions. Instead of shifting the onsets, the accents can receive an equivalent shift, while the onsets themselves are kept in their initial positions. In that way, the accent of a loud event in a strong position will be anticipated and will be shifted to a preceding event in a weaker metrical position.

In order for the shift to be effective, the amplitude of the onset in the strong position must be attenuated. Looking closer at the metrical template and the way it is constructed, one observes that strong and weak pulses are always alternating. Strong pulses, which belong to slow metrical levels, are always followed by weak pulses, which belong to faster metrical levels. Shifting the metrical template to the left by one pulse provides us the attenuated amplitude for the strong event (Fig. 13D, solid arrows).

The generation of syncopation based on dynamic accents is a secondary mechanism that is needed only when the onsets themselves cannot be shifted. Otherwise, the onsets carry with them the dynamic accents, so that onsets and accents are anticipated together.

The example in Fig. 13 illustrates how dynamic accents are used. Metrical accents are calculated according to the metrical template (A). They are, then, assigned to the onsets of a de-syncopated pattern (B and C). Finally, syncopation is generated in two ways (D): (1) by shifting the onsets that also carry their accents, and (2) by shifting the metrical accents instead of the onsets themselves.

## 6 Max4Live Device

Our MIDI device manipulates effectively and in real time the syncopation in midi clips. It systematically removes the syncopation and creates new syncopation in the clip according to the transformations described in Sects. 4 and 5.

The device constructs automatically a metrical template that corresponds to the time signature and tempo settings of the Ableton Live Set. The template defines the fastest metrical subdivision, i.e. a quantization grid, and the number of pulses or steps in the pattern as described in Sect. 3.

The user interface of the device is shown in Fig. 14. When a midi clip is playing in the same track as the device, its contents are automatically “loaded” in the Max4Live device. The midi notes are grouped together according to the pulse they belong. In the center of the device, a display area presents a rough representation of the midi contents to the user. Pulses containing note events are shown as red squares. Grey bars represent the “beat”, i.e. the pulses that belong to the slowest metrical level of the template.

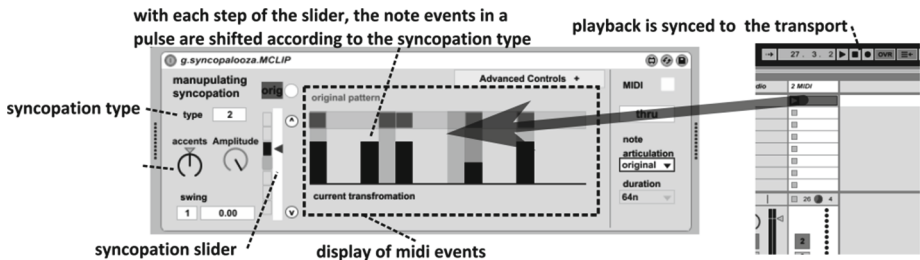


Fig. 14. The MIDI Max4Live device that manipulates the syncopation in midi clips.

The syncopation transformations are applied on each pulse as if it was a binary pattern. Pulses containing note events are treated as the onsets of the binary pattern. When an onset is displaced to another pulse, all the note events of the respective pulse are shifted together. The transformations are applied automatically according to the algorithms described in Sect. 4.3. The pattern is first de-syncopated and its root is determined. Then a branch is generated starting from the root and reaching an end pattern. The type of the “extra” transformations needed to syncopate the midi notes beyond the original syncopation can be set on the left side of the device through a number box named “type”.

A syncopation slider on the left of the display area controls which pattern in the generated branch is selected for playback. At the lower end of the slider, the root pattern is found. As one raises the slider, he will pass by the original midi clip that is marked as a red square on the slider. Raising the slider further will lead to further syncopation until the end of the branch is reached at the upper end of the slider.

A different branch can be generated that does not pass by the original pattern in the midi clip and does not necessarily end at the same end pattern. By de-activating the red button named “orig” right above the slider, a branch is generated based on a default order and the user input type value. The order is chosen according to the indispensability values of Barlow’s algorithm [1, 2].

For the onsets that cannot be shifted, their dynamic accents are shifted as described in Sect. 5. First, dynamic accents are imposed on the note events, substituting their original velocity values with velocity values that match the metrical template. Second, when a specific transformation in the branch array cannot be applied (because a note event “blocks” it), the velocities are shifted instead. A dial at the left side of the device controls the dynamic accents applied. When the dial is turned to the left, no accents are applied and all onsets have maximum velocity. When the dial is turned fully right, only the slower metrical level survives. In the center position, the velocity of the events is relative to the metrical strength of their positions.

In the display area and just below the red squares denoting the midi clip contents, a representation of the currently selected pattern is shown. As one moves the syncopation slider, the pattern that is selected changes and it is displayed as black bars that represent the pulses containing note events. The height of each bar is relative to the midi velocity assigned to the note events in the pulse.

A few other secondary controls are available: (i) for choosing the articulation of the midi events (staccato, legato, original durations or portamento), (ii) for controlling the global amplitude of the output of the device, and (iii) for choosing the midi thru operation (blocking or letting through the midi notes that come in the midi input of the device).

## 7 Conclusions

In this paper we presented a set of formalized generic transformations that can analyze, generate and manipulate the syncopation in binary patterns. The transformations are based on the cognitive definition of syncopation as a violation of metrical expectancies. The transformations serve as a tool for analyzing and generating syncopation in

rhythmic patterns. In fact the two processes are combined under the concept of a syncopation transformation; the transformation describes the syncopation in a pattern at the same time that it generates the pattern itself. We extended the main transformations to include dynamic accents that can be applied on binary patterns to enhance the metrical feel or generate syncopation.

We introduce the concept of the syncopation tree. The rhythmic patterns found on the branches of the tree are interconnected through specific sequences of transformations. Any two patterns on a tree can be transformed from one to the other following a branch of single-step transformations. One can navigate from one pattern to another unveiling the relations between the patterns. In each tree, it exists only one pattern with no syncopation, called the root pattern. The particular branch that connects a pattern to its root provides with a detailed description of the syncopation in the pattern, such as the number of syncopating events, the position of the syncopations and the metrical levels involved. The transformations and syncopation tree can serve as the basis for rhythmic similarity or distance measures.

The way syncopation is encoded and described in this model helps in creatively exploring its uses. For example, one can think of modeling the syncopation of one particular music style as specific transformations on certain root patterns. These transformations could then be applied on root patterns that do not belong to the modeled style. The modeling of a musical style can be done by automatically de-syncopating a number of patterns characteristic of that style, e.g. rhythmic patterns performed simultaneously by different instruments. The resulted transformations could then be combined and applied to the root of any other pattern effectively “copying” the syncopation style.

The applications of the syncopation transformations to music and syncopation styles are left to be explored in the future work of our group. However, we developed a software application as a small example of a creative application of the syncopation transformations. It has the form of a Max4Live midi device that manipulates the syncopation of midi clips.

The Max4Live devices and related externals are available for download at our website: <http://smc.inescporto.pt/shakeit/>.

**Acknowledgments.** This research was partly funded by the Media Arts and Technologies project (MAT), NORTE-07-0124-FEDER-000061, financed by the North Portugal Regional Operational Programme (ON.2 – O Novo Norte), under the National Strategic Reference Framework (NSRF), through the European Regional Development Fund (ERDF), and by national funds, through the Portuguese funding agency, Fundação para a Ciência e a Tecnologia (FCT).

## References

1. Barlow, C.: Corrections for Clarence Barlow’s article: two essays on theory. *Comput. Music J.* **11**(4), 10 (1987)
2. Barlow, C., Lohner, H.: Two essays on theory. *Comput. Music J.* **11**(1), 44–60 (1987)
3. Fitch, W.T., Rosenfeld, A.J.: Perception and production of syncopated rhythms. *Music Percept.* **25**(1), 43–58 (2007)

4. Gabrielsson, A.: Adjective ratings and dimension analyses of auditory rhythm patterns. *Scand. J. Psychol.* **14**(4), 244–260 (1973)
5. Gómez, F., Thul, E., Toussaint, G.: An experimental comparison of formal measures of rhythmic syncopation. In: *Proceedings of the International Computer Music Conference*, pp. 101–104 (2007)
6. Gómez, F., Melvin, A., Escuela, U.: Mathematical measures of syncopation. In: *Proceedings of BRIDGES: Mathematical Connections in Art, Music and Science*, pp. 73–84 (2005)
7. Huron, D.: *Sweet Anticipation: Music and the Psychology of Expectation*. The MIT Press, Cambridge (2006)
8. Huron, D., Ommen, A.: An empirical study of syncopation in American popular music, 1890–1939. *Music Theory Spectr.* **28**(2), 211–231 (2006)
9. Jones, M.R.: Musical time. In: Hallam, S., et al. (eds.) *The Oxford Handbook of Music Psychology*, pp. 81–92. Oxford University Press, Oxford (2009)
10. Keith, M.: *From Polychords to Polya: Adventures in Musical Combinatorics*. Vinculum Press, Princeton (1991)
11. Kennedy, M., Bourne, J. (eds.): *Oxford Dictionary of Music*. Oxford University Press, New York (1994)
12. Lerdahl, F., Jackendoff, R.: *A Generative Theory of Tonal Music*. The MIT Press, Cambridge (1983)
13. London, J.: *Hearing in Time*. Oxford University Press, Oxford (2012)
14. Longuet-Higgins, H.C., Lee, C.S.: The rhythmic interpretation of monophonic music. *Music Percept.* **1**(4), 424–441 (1984)
15. Miron, M., Davies, M., Gouyon, F.: An open-source drum transcription system for pure data and max MSP. In: *The 38th International Conference on Acoustics, Speech, and Signal Processing*, Vancouver, Canada (2013)
16. Palmer, C., Krumhansl, C.L.: Mental representations for musical meter. *J. Exp. Psychol.* **16**(4), 728–741 (1990)
17. Parncutt, R.: A perceptual model of pulse salience and metrical accent in musical rhythms. *Music Percept.* **11**(4), 409–464 (1994)
18. Randel, D.M.: *The Harvard Dictionary of Music*. Belknap Press of Harvard University Press, Cambridge (1986)
19. Repp, B.H.: Rate limits of sensorimotor synchronization. *Adv. Cogn. Psychol.* **2**(2–3), 163–181 (2006)
20. Sioros, G., Miron, M., Cocharro, D., Guedes, G., Gouyon, F.: Syncopalooza: manipulating the syncopation in rhythmic performances. In: *Proceedings of the 10th International Symposium on Computer Music Multidisciplinary Research*, pp. 454–469. Laboratoire de Mécanique et d'Acoustique, Marseille (2013)
21. Sioros, G., Guedes, C.: A formal approach for high-level automatic rhythm generation. In: *Proceedings of the BRIDGES 2011 – Mathematics, Music, Art, Architecture, Culture Conference*, Coimbra, Portugal (2011)
22. Sioros, G., Guedes, C.: Complexity driven recombination of MIDI loops. In: *Proceedings of the 12th International Society for Music Information Retrieval Conference*, Miami, Florida, USA, pp. 381–386 (2011)
23. Temperley, D.: Syncopation in rock: a perceptual perspective. *Pop. Music.* **18**(1), 19–40 (1999)
24. Volk, A., de Haas, W.: A corpus-based study on ragtime syncopation. In: *14th International Society for Music Information Retrieval Conference*, Curitiba, Brazil (2013)
25. Yeston, M.: *The Stratification of Musical Rhythm*. Yale University Press, New Haven (1976)