

Język L-M jako struktura danych

1. Wstęp

Prezentujemy tutaj jeden z możliwych sposobów przekształcenia muzyki w języku L-M na strukturę danych. Struktura ta składa się z podstawowych typów obiektów, powszechnie używanych w wielu językach programowania, toteż pozostaje praktycznie niezależna od środowiska przetwarzania. Do opisu struktury danych używamy składni języka JSON – prostego i uznanego formatu zapisu danych.

Opisywana struktura jest w zamyśle platformą wymiany informacji pomiędzy różnymi notacjami muzycznymi. Warstwa informacji czysto muzycznej jest niezależna od żadnej konkretnej notacji. Trudno jednak o całkowite odseparowanie informacji muzycznej od notacyjnej. Struktura danych opisująca wyłącznie wysokości dźwięków i czas ich trwania, bez żadnych dodatkowych elementów dotyczących formy zapisu, byłaby w praktyce mało przydatna. Wynika to z faktu, że żadna z istniejących notacji muzycznych, także standardowa notacja czarnodrukowa, nie jest jednoznaczna. Tę samą muzykę można zapisać na wiele różnych sposobów w obrębie jednej tylko notacji. Jeśli struktura danych ma służyć za praktyczne medium wymiany informacji muzycznej i umożliwiać sensowną translację między różnymi notacjami, musi uwzględniać także elementy notacyjne.

2. Struktura

2.1. Element nadrzędny

Nadrzędnym elementem struktury jest słownik takiej postaci:

```
{
  "parts" : [ ... ],
  "stats" : { ... },
}
```

Wartość klucza `parts` to lista partytur składających się na utwór muzyczny. Pod kluczem `stats` zawarty jest słownik parametrów pomocniczych, mających wpływ na interpretację innych elementów struktury. Najważniejszym z tych parametrów jest `time_scale`. Jest to względna skala czasu trwania elementów struktury muzycznej (taktów, głosów, nut), wyrażona liczbą całkowitą.

```
{
  "parts" : [ ... ],
  "stats" : {
    "time_scale" : 512,
    "time_grid": 32,
    "time_div": 108
  }
}
```

W tym przypadku liczba 512 oznacza, że w całej strukturze wartości nutowe wyrażone są w pewnych jednostkach względnych, takich, że 512 oznacza czas trwania całej nuty. Dobór tej wielkości jest arbitralny, ale musi uwzględniać najdrobniejsze wartości nutowe (wraz z ewentualnymi następującymi po nutach kropkami), aby wszystkie czasy trwania w całym utworze muzycznym dawały się zapisać w spójny sposób. Inne parametry zapisane w słowniku `stats` ułatwiają wyliczanie miar czasu także dla nut zawartych w grupach nieregularnych (triol, kwintol, septymol itd.).

2.2. Partytury

Główną zawartość struktury powstałej z przekształcenia języka L-M stanowi obiekt zapisany w kluczu `parts` nadrzędnego słownika. Klucz `parts` jest zawsze obecny, nawet jeśli utwór muzyczny nie posiada jawnego podziału na partytury. W takim przypadku lista partytur zawiera tylko jeden element, opisujący jedyną partyturę utworu (na przykład partię głosu albo fortepianu). To samo podejście stosujemy do wszystkich elementów struktury; jeśli pewien element może wystąpić w strukturze więcej niż jeden raz, reprezentowany jest zawsze jako lista, nawet jeśli w większości przypadków będzie to lista z jednym tylko elementem. Dzięki temu każdy element struktury posiada jeden konkretny typ (na przykład zawsze słownik albo zawsze lista). Wyjątek stanowią jedynie elementy opcjonalne, których wartością może być `null`. Każdy element występuje na określonym, niezmiennym poziomie zagnieżdżenia, niezależnie od złożoności utworu muzycznego.

Każdy element listy `parts` to słownik następującej postaci:

```
{
  "staves" : [ ... ],
  "name" : "P1"
}
```

Pod kluczem `staves` znajduje się lista obiektów opisujących pięciolinie danej partytury. Klucz `name` zawiera symboliczną nazwę partytury. Identyfikator partytury może być dowolny, musi być jednak unikalny dla całej struktury.

2.3. Pięciolinie

Zestaw pięciolinii to lista takiej postaci:

```
[
  { ... staff1 ... },
  { ... staff2 ... },
  ...
]
```

Każdy element tej listy to słownik:

```
{
  "measures" : [ ... ],
  "number" : 1,
  "name" : "right_hand"
}
```

Pod kluczem `measures` znajduje się lista obiektów opisujących takty w danej pięciolinii. Klucz `number` określa kolejność pięciolinii w tradycyjnym zapisie nutowym. `name` to opisowy identyfikator pięciolinii, na przykład `right_hand` oraz `left_hand` dla prawej i lewej ręki muzyki fortepianowej.

2.4. Takty

Sekwencja taktów w obrębie pięciolinii to lista:

```
[
  { ... measure1 ... },
  { ... measure2 ... },
  ...
]
```

Każdy element tej listy to słownik opisujący takt:

```
{
  "number": 1,
  "metrum": { ... },
  "bar": { ... },
  "mood": { ... },
  "key": { ... },
  "clef": { ... },
  "voices": [ ... ]
}
```

Wartość `number` to numer taktu. `metrum` zawiera obiekt charakteryzujący metrum dla danego taktu. `bar` to słownik opisujący położenie taktu w większej strukturze muzycznej, na przykład, czy jest to pierwszy takt pewnej powtarzanej części utworu lub ostatni takt przed powtórzeniem. `mood` to obiekt przechowujący informację o dodatkowych instrukcjach muzycznych, takich jak dynamika i tempo. `key` to obiekt definiujący tonację, w której grany jest dany takt. Obiekt pod kluczem `clef` zawiera informację o zastosowanych kluczach muzycznych. Główna zawartość muzyczna taktu przechowywana jest w obiekcie pod kluczem `voices`.

2.5. Metrum

Metrum taktu w strukturze L-M zapisane jest jako słownik.

```
{
  "beat": 8,
  "beats": 3,
  "implied": false
}
```

Wartość `beats` określa liczbę jednostek rytmicznych, natomiast wartość `beat` miarę jednostki rytmicznej. Wartość logiczna `implied` informuje o tym, czy metrum jest jawnie podane na początku taktu (`false`), czy też wynika z metrum podanego wcześniej, na przykład na początku utworu (`true`). Powyższy przykład opisuje metrum $3/8$, odziedziczone przez bieżący takt, a podane w którymś z wcześniejszych taktów.

2.6. Powtórzenia

Umieszczenie taktu w większej strukturze muzycznej (na przykład powtarzanym fragmencie utworu) definiuje słownik pod kluczem `bar`:

```
{
  "left" : left_bar,
  "right" : right_bar
}
```

Symbole `left_bar` i `right_bar` mogą przyjmować wartości nawiązujące do rodzaju kresek taktowych używanych w standardowej notacji muzycznej do oznaczenia powtórzeń:

```
> "measure"
> "section"
> "repeat"
> "forward"
> "end"
```

Oba klucze `left` i `right` mogą mieć wartość `null`.

2.7. Mood

Dynamikę taktu opisuje słownik

```
{
  "tempo": 150,
  "text": "Allegro"
}
```

Klucz `tempo` określa tempo taktu w dowolnej względnej jednostce, spójnej dla całego utworu. Można na przykład przyjąć jednostki czasu stosowane w formacie MIDI. Klucz `text` zawiera dowolny tekst użyty do opisowego określenia dynamiki, na przykład "Allegro" albo "Largo".

2.8. Tonacja

Tonacja opisana jest słownikiem:

```
{
  "fifths": 0,
  "implied": false
}
```

Klucz `fifths` określa liczbę znaków przykluczowych, a tym samym pozycję na kole kwintowym. Jeśli wartość `fifths` jest dodatnia, należy ją traktować jako liczbę krzyżyków użytych przy kluczu do oznaczenia tonacji. Jeśli wartość `fifths` jest ujemna, jej bezwzględna wartość to liczba bemoli. Wartość zero określa tonację *C-dur* lub *a-moll*, które w zapisie nutowym nie mają żadnych znaków przy kluczu. Wartość `implied` informuje o tym, czy tonacja (ew. zmiana tonacji) została jawnie podana w bieżącym takcie (wartość `false`), czy też wynika z tonacji określonej w poprzednich taktach, na przykład na początku utworu (wartość `true`).

2.9. Klucz

Znak użytego klucza muzycznego opisuje słownik:

```
{
  "type": "treble",
  "implied": true
}
```

Wartość `type` określa symboliczną nazwę klucza, na przykład:

```
▷ "treble"
▷ "alto"
▷ "tenor"
▷ "bass"
```

Wartość `implied` informuje o tym, czy klucz został jawnie podany w obrębie danego taktu (wartość `false`), czy też został podany wcześniej, na przykład na początku utworu (wartość `true`).

2.10. Głosy

Głosy w obrębie taktu w strukturze L-M reprezentowane są jako lista postaci:

```
[
  { ... voice1 ... },
  { ... voice2 ... },
  ...
]
```

Każdy z elementów tej listy to słownik charakteryzujący głos:

```
"number" : 1,
"start" : 0,
"end" : 192,
"time" : 192,
"notes" : [ ... ]
```

`number` to numer głosu. Numeracja głosów jest arbitralna, przy czym numery głosów traktowane są jako ich identyfikatory, muszą więc pozostawać unikalne dla całego taktu. Wartość `start` określa czas początku głosu względem początku taktu (zero oznacza rozpoczęcie głosu wraz z początkiem taktu). Wartość `end` oznacza koniec trwania głosu. Wartość `time` to suma wartości nutowych nut i pauz składających się na dany głos. Jeśli struktura L-M opisuje poprawny rytmicznie utwór, wartość `time` jest równa różnicy `end` i `start`. Wszystkie trzy wartości `start`, `end` i `time` wyrażone są we względnych jednostkach czasu, podanych pomocniczym słowniku `stats` w nadrzędnym elemencie struktury.

Jeśli za skalę czasu dla całego utworu przyjęto wartość 512, wartość 192 oznacza $\frac{3}{8}$ czasu trwania całej nuty, czyli łączny czas trwania trzech wartości ósemkowych. Jeśli metrum taktu to $\frac{3}{8}$, czas trwania przykładowego głosu jest taki sam, jak czas trwania całego taktu. Głos trwający tyle samo co cały takt to najbardziej typowy przypadek. W strukturze L-M można jednak opisać bardziej ogólny przypadek, w którym pewien głos nie zaczyna się w wraz z początkiem taktu i nie kończy się wraz z końcem taktu. Takie pojęcie głosu można spotkać w BRAILLE-owskiej notacji muzycznej.

2.11. Nuty

Nuty w obrębie głosu w strukturze L-M reprezentuje lista:

```
[
  { ... note1 ... },
  { ... note2 ... },
  ...
]
```

Każdy z elementów tej listy to słownik opisujący pojedynczą nutę lub zespół nut zagranych w tym samym momencie i trwających przez taki sam czas, czyli akord.

```
{
  "tones" : [ ... ],
  "value" : 16,
  "time" : 32,
  "dots" : 0,
  "beam" : beam_spec,
  "tie" : tie_spec,
  "slur" : slur_spec,
  "art" : [ ... ]
}
```

Wartość `tones` to lista pojedynczych dźwięków. W przypadku pojedynczej nuty jest to lista jednoelementowa, w przypadku akordu, lista będzie miała kilka elementów. Pod kluczem `value` mieści się wartość nutowa; 1 oznacza całą nutę, 2 oznacza półnutę, 4 ćwierćnutę itd. Klucz `time` zawiera czas trwania nuty, wyrażony w opisywanych wcześniej jednostkach względnych. W strukturze opisującej poprawny notacyjnie utwór muzyczny, wartości `value`, `time` oraz przyjęta dla utworu skala czasu `time_scale`, pozostają w ścisłej relacji: $time_scale = value * time$. Dodatkową informacją dotyczącą czasu trwania pewnej nuty jest wartość `dots`, określająca liczbę kropek przynutowych w standardowej notacji muzycznej. Brak takiego klucza w strukturze opisującej nutę jest równoznaczny wartości zero. Słownik reprezentujący nutę może także zawierać informację o belkach, łukach i symbolach artykulacyjnych; klucze `beam`, `tie`, `slur` i `art` opisane są w kolejnych sekcjach. Każdy z nich jest opcjonalny – brak takiego klucza w słowniku (lub wartość `null`) jest równoznaczna z brakiem danej cechy.

Podobna struktura używana jest do opisania pauz, przy czym wszystkie wartości charakterystyczne dla nut, w szczególności `tones`, są wtedy pominięte. Traktujemy pauzę jako specyficzną „niebrzmiającą” nutę, nie posiadającą powiązanego dźwięku. Oto przykład struktury reprezentującej pauzę:

```
{
  "value" : 16,
  "time" : 32
}
```

2.12. Tony

Zestaw tonów składających się na akord to lista:

```
[
  { ... tone1 ... },
  { ... tone2 ... },
  ...
]
```

Każdy element listy `tones` to słownik następującej postaci:

```
{
  "pitch": "e",
  "octave": -1,
  "accidental" : { ... }
}
```

Wartość `pitch` określa wysokość dźwięku w siedmiostopniowej skali od *c* do *h*. Wartość `octave` definiuje oktawę, w której grany jest dźwięk. `accidental` to modyfikator wysokości dźwięku odzwierciedlający zastosowanie znaków chromatycznych.

2.13. Wysokość dźwięku

Możliwe wartości parametru `pitch` to

- ▷ "c"
- ▷ "d"
- ▷ "e"
- ▷ "f"
- ▷ "g"
- ▷ "a"
- ▷ "h"

2.14. Oktawy

Dla oktaw przyjęliśmy oznaczenia numeryczne, stosowane także w formacie MIDI:

- ▷ -3 – subkontra
- ▷ -2 – kontra
- ▷ -1 – wielka
- ▷ 0 – mała
- ▷ 1 – jednokreślna
- ▷ 2 – dwukreślna
- ▷ 3 – trzykreślna
- ▷ 4 – czterokreślna
- ▷ 5 – pięciokreślna

2.15. Znaki chromatyczne

Wysokość dźwięku wynikająca z parametrów `pitch` i `octave` może być zmodyfikowana poprzez obecność obiektu `accidental`. Obiekt ten ma następującą formę:

```
{
  "alter": 1,
  "implied": false
}
```

Klucz `alter` definiuje wielkość modyfikacji dźwięku podaną jako liczbę półtonów. Wartość dodatnia oznacza modyfikację w górę, wartość ujemna modyfikację w dół. Wartość zero oznacza kasownik. Wartość absolutna parametru `alter` nigdy nie jest większa niż 2. Dodatkowy parametr `implied` informuje o tym, czy znak chromatyczny został użyty przy danej nucie w sposób jawny (poprzez wstawienie znaku krzyżyka lub bemola), czy też wynika z tonacji utworu i jest niejako dziedziczony ze znaków chromatycznych umieszczonych przy kluczu, w tym samym lub poprzednich taktach.

2.16. Łuki

Obecność łuków rytmicznych łączących sąsiednie nuty oznaczona jest w strukturze L-M dodatkowym kluczem `tie` w słowniku reprezentującym nutę. Wartością tego klucza jest słownik

```
{
  "start" : tie_spec,
  "end" : tie_spec
}
```

`tie_spec` to wartość logiczna, określająca czy nuta rozpoczyna łuk (`start` ma wartość `true`), czy go kończy (`end` ma wartość `true`). Brak jednego z kluczy `start` lub `end` jest jednoznaczny z wartością `false` lub `null`. Specyficzny przypadek, w którym dana nuta kończy jeden łuk i jednocześnie rozpoczyna następny, byłaby zapisana w taki sposób:

```
{
  "start" : true,
  "end" : true
}
```

2.17. Ligatury

Obecność łuków łączących ciąg nut w pewne frazy (ligatur) oznaczona jest dodatkowym kluczem `slur` w słowniku nuty. Wartością tego klucza jest słownik:

```
{
  "start" : slur_spec,
  "continue" : slur_spec,
  "end" : slur_spec
}
```

W przeciwieństwie do łuków rytmicznych, ligatury mogą być zagnieżdżone, to znaczy jedna nuta może należeć do więcej niż jednej ligatury. Aby odzwierciedlić to w strukturze, wartość `slur_spec` to lista numerycznych identyfikatorów łuków. Na przykład, nuta należąca do ligatury o identyfikatorze 1, kończąca ligaturę o identyfikatorze 2 i rozpoczynająca ligaturę o identyfikatorze 3, opisana byłaby następująco:

```
{
  "start" : [ 3 ],
  "continue" : [ 1 ],
  "end" : [ 2 ]
}
```

2.18. Grupy regularne

Obecność belek łączących nuty reprezentowana jest w strukturze L-M jako dodatkowy klucz `beam` w słowniku nuty. Wartość `beam` może mieć następujące wartości:

```
> "start"
> "continue"
> "end"
```

Wartość `start` oznacza, że nuta jest pierwszą nutą grupy nut pod belką. Wartość `continue` oznacza, że nuta jest kolejną (nie pierwszą i nie ostatnią) nutą grupy. Wartość `end` oznacza ostatnią nutę grupy.

2.19. Grupy nieregularne

Jeśli nuta należy do grupy nieregularnej (na przykład trioli), w jej słowniku znajduje się dodatkowy klucz `tuple`, określający położenie nuty w obrębie tej grupy. Wartością klucza `tuple` jest słownik:

```
{
  "no": 1,
  "of": 3,
  "normal_time": 192,
  "actual_time": 128
}
```

Wartość klucza `no` to pozycja nuty w grupie nieregularnej, natomiast wartość `of` oznacza całkowitą liczbę nut w grupie (w tym przykładzie jest to pierwsza nuta z trzech). Wartości `normal_time` oraz `actual_time` określają współczynnik o jaki zmienia się czas trwania nut w obrębie grupy. W powyższym przykładzie ten współczynnik to 2/3 (trzy nuty w trioli ósemkowej zagrane w czasie dwóch ósemek).

2.20. Akcenty i symbole artykulacyjne

Informacja o dodatkowych symbolach stosowanych przy nutach, w szczególności o akcentach i znakach artykulacyjnych, może być zawarta w słowniku nuty pod kluczem `art`. Elementami tej listy są opisowe identyfikatory symboli, na przykład:

```
> "stacatto"
> "accent"
> "tenuto"
> "martellato"
> "stacatissimo"
> "swell"
> "bow_up"
> "bow_down"
> "fermata"
```

- ▷ "full_breatch"
- ▷ "half_breath"
- ▷ "pedaling_down"
- ▷ "pedaling_half"
- ▷ "pedaling_up"

Struktura L-M nie ogranicza zbioru dopuszczalnych symboli i nie narzuca sposobu ich zapisu.