

## PROJECT 1 - 6.005 ABC Player

### Datatypes

```
Pitch = Pitch(value: int,accidental: int,octave: int)
NoteLength = NoteLength(numerator: int,denominator: int)
NoteOrRest = Note(p: Pitch,l: NoteLength) + Rest(l: NoteLength)
Tuplet = Duplet(a: Note,b: Note) + Triplet(a: Note,b: Note,c: Note) + Quadruplet(a:
Note,b: Note,c: Note,d: Note)
Chord = Chord(notes: ImList<Note>)
PlayableElement = Chord + Tuplet + NoteOrRest
Measure = RegularMeasure(elements: ImList<PlayableElement>,boolean:
isMajorStart,measureID: int,nextMeasureID: int) + RepeatableMeasure(elements:
ImList<PlayableElement>,boolean: isMajorStart,measureID: int,nextMeasureID:
int,repeatMeasureID: int) + IncompleteMeasure(elements:
ImList<PlayableElement>,measureID: int,nextMeasureID: int)
Voice = Voice(name: String, music: List<Measure>, playingOrder: List<int>)
```

MeterTempo = MeterTempo(numerator: int, denominator: int, bpm: int, ticksPerUnit: int, defaultNoteNumerator: int, defaultNoteDenominator: int) //I added Note in the names of the default note length values (for clarity)

AbcHeader = AbcHeader(title: String, pieceNumber: int, composer: String, mtempo: MeterTempo, key: Key)

Key = Key(modifiers: int) //modifiers is a bitmask containing information about sharps and flats for all notes: binary representation - xbagfedc - x is 1 if sharps, 0 if flats; c is 1 if note C has a modifier; d is 1 if note D has a modifier and so on. Will employ lookup table to translate between key names (e.g. "Cbm") to bitmasks.

KeyTempoChange(firstMeasureID: int,mtempo: MeterTempo,k: Key) //datatype that specifies the measure when a new tempo/key is specified in the middle of the file; assumes that measure IDs are consistent across voices (the first measure in every voice is measure ID 0, the second has ID 1 and so on).

AbcMusic = AbcMusic(changes: KeyTempoChange,voices: ImList<Voice>)

AbcFile = AbcFile(header: AbcHeader,music: AbcMusic)

**Team:**

Jesika H Haria  
 Kaivan Wadia  
 Predrag Gruevski

**ABC Grammar**

*A subset of ABC 1.6 in BNF format for 6.005 Project 1*

```

abc-file ::= abc-header abc-music
abc-header ::= field-number comment* field-title other-fields* field-key
field-title ::= "T:" text end-of-line
other-fields ::= field-composer | field-default-length | field-meter
               | field-tempo | field-voice | comment | field-index
field-composer ::= "C:" text end-of-line
field-default-length ::= "L:" note-length-strict end-of-line
field-meter ::= "M:" meter end-of-line
field-tempo ::= "Q:" tempo end-of-line
field-voice ::= "V:" text end-of-line
field-key ::= "K:" key end-of-line
field-index ::= "X:" key end-of-line

key ::= "C" | "G" | "D" | "A" | "E" | "B" | "F#" | "C#" | "F" | "Bb" | "Eb"
       | "Ab" | "Db" | "Gb" | "Cb" | "Em" | "Bm" | "F#m" | "C#m" | "G#m" | "D#m"
       | "A#m" | "Dm" | "Gm" | "Cm" | "Fm" | "Bbm" | "Ebm" | "Abm"

meter ::= "C" | "C|" | meter-fraction
meter-fraction ::= DIGIT+ "/" DIGIT+

tempo ::= DIGIT+

;;;;;;;;;; END OF HEADER ;;;;;;;;;;

abc-music ::= abc-line+
abc-line ::= (measure+ end-of-line) | mid-tune-field | comment
measure ::= [space+] [nth-repeat] [space+] playable-element+ [space+] barline

playable-element ::= note | chord | tuplelet-element

; note is either a pitch or a rest
note ::= note-or-rest [note-length]
note-or-rest ::= pitch | rest
pitch ::= [accidental] basenote [octave]
octave ::= ("'"+" ) | (" , "+" )
note-length ::= [DIGIT+] ["/" [DIGIT+]]
note-length-strict ::= DIGIT+ "/" DIGIT+
; "^" is sharp, "_" is flat, and "=" is neutral
accidental ::= "^" | "^^" | "_" | "___" | "="

basenote ::= "C" | "D" | "E" | "F" | "G" | "A" | "B"
             | "c" | "d" | "e" | "f" | "g" | "a" | "b"

rest ::= "z"

; tuplelets
tuplelet-element ::= tuplelet-spec (note | chord)+
tuplelet-spec ::= "(" DIGIT
; chords
chord ::= "[" note+ "]"

```

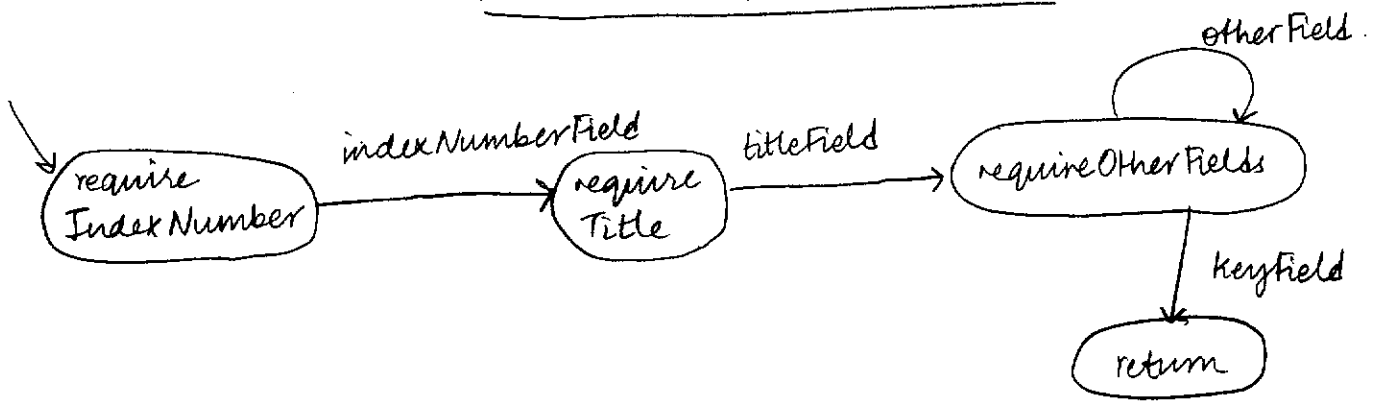
**Team:**

Jesika H Haria  
Kaivan Wadia  
Predrag Gruevski

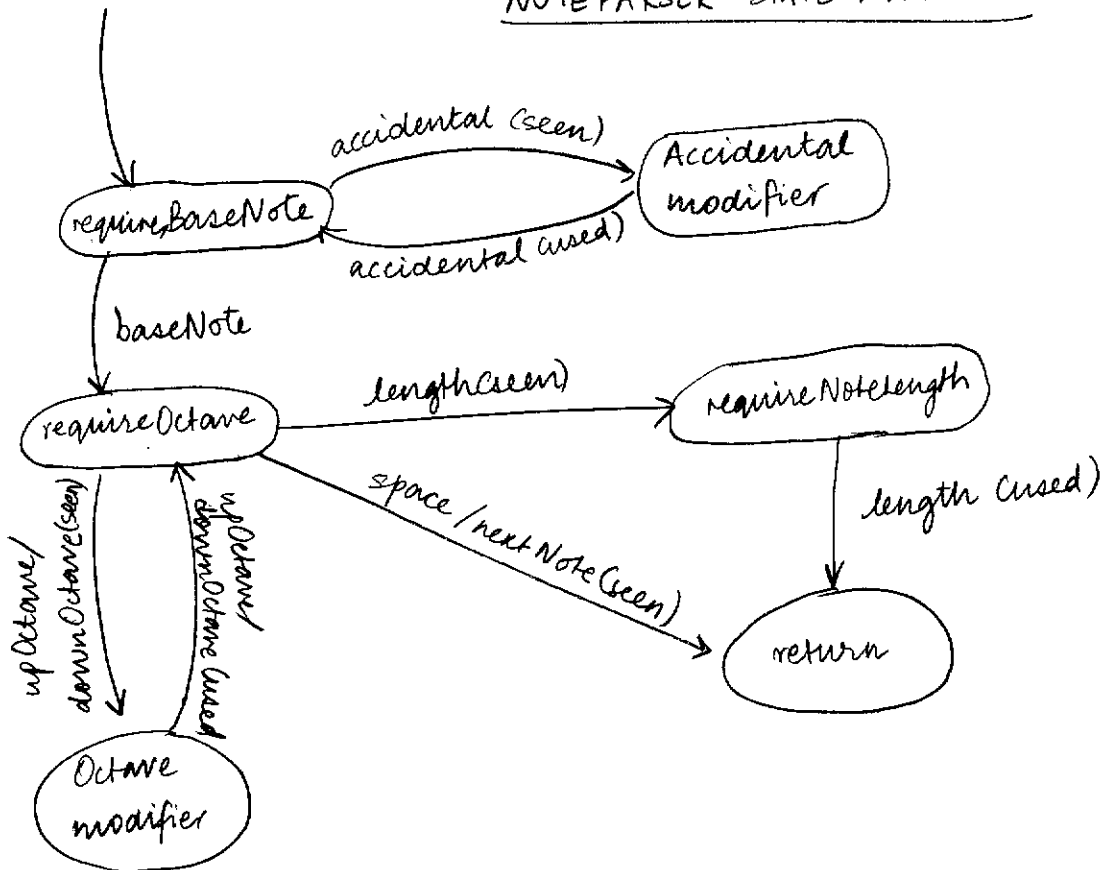
```
barline ::= "|" | "||" | "[" | "]" | ":" | ";"  
nth-repeat ::= "[1" | "[2"
```

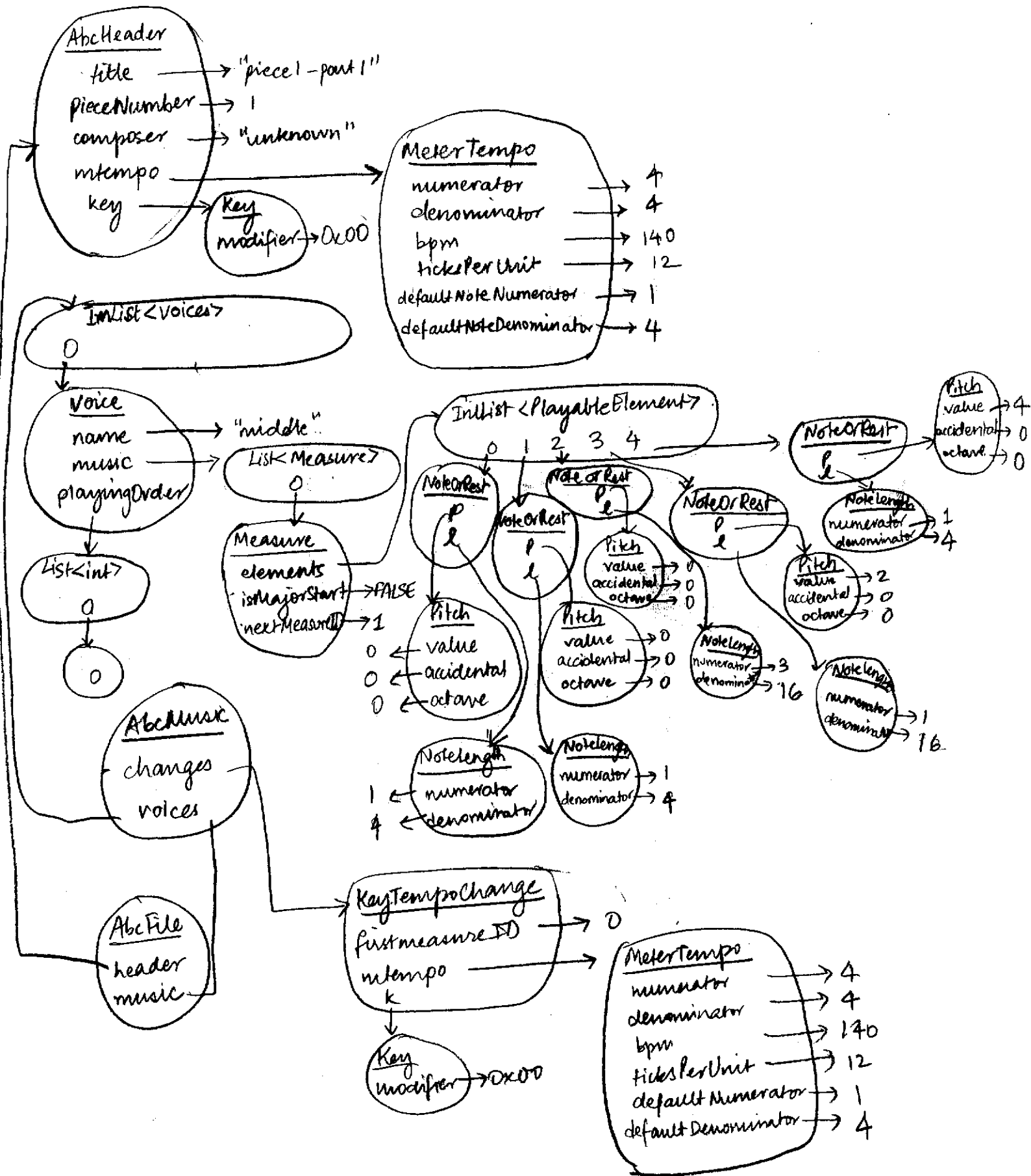
```
;;;;;;;;;;;;; MISC ;;;;;;;;;;;;;;  
mid-tune-field- ::= field-voice | field-tempo  
comment ::= "%" text linefeed  
end-of-line ::= comment | linefeed
```

## HEADER PARSER STATE MACHINE



## NOTE PARSER STATE MACHINE





ABC FILE  
header  
music

ABC HEADER

title  
piece Number  
mtempo  
key

66 chords

2256

Key  
modifier

0x00

METER TEMPO

numerator → 4  
denominator → 4  
bpm → 140  
ticks per Unit → 12  
default Note Numerator → 1  
default Note Denominator → 4

ABC MUSIC  
changes  
voices

Key Tempo Change  
first measure ID → 0  
mtempo →

Meter Tempo

numerator → 4  
denominator → 4  
bpm → 140  
ticks per Unit → 12  
default Note Numerator → 1  
default Note Denominator → 4

ImList <voices>

VOICE

name  
music  
playing Order

66 lower

List <measure>  
0 1

MEASURE

elements  
is major start → FALSE  
next measure ID → 1

List <int>  
0 0 1

Measure  
elements  
is major start → FALSE  
next measure ID → 1

ImList <Playable Elements>  
0 1

ImList <Playable Elements>  
0 1

NOTE  
P  
L

NOTE  
P  
L

NOTE  
P  
L

CHORD  
notes

ImList <NOTE>  
0 1

NOTE LENGTH  
num → 3  
den → 4

NOTE LENGTH  
num → 1  
den → 2

NOTE LENGTH  
num → 1  
den → 2

NOTE  
P  
L

PITCH  
val → 2  
acc → 0  
oct → 0

NOTE  
P  
L

PITCH  
val → 9  
acc → 0  
oct → 1

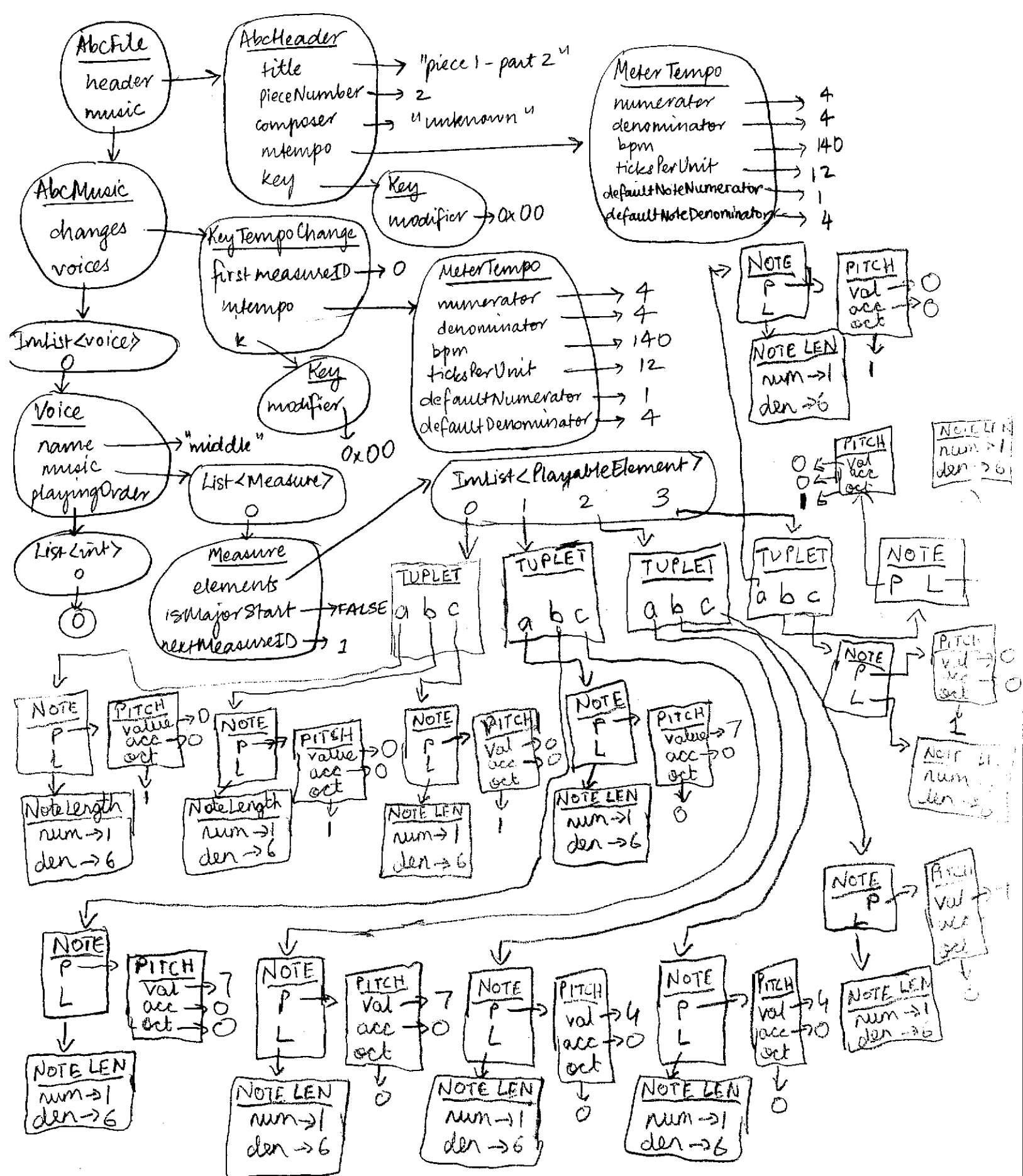
PITCH  
val → 2  
acc → 1  
oct → c

NOTE LENGTH  
num → 1  
den → 4

NOTE LENGTH  
num → 1  
den → 4

PITCH  
val → 4  
acc → 0  
oct → 0

[1:[ED]c3:1 a2#D2]



(3c/c/c) (3G/G/G) (3E/E/E) (3Y/Y/Y)

