Will Schell
MKR CFG Exercises

CFG for ABC Notation Notes and Rests:

Terminals: {‘ ^ , _ < > / = A B C D E F G X Z a b c d e f g z 0 1 2 3 4 5 6 7 8 9}

Non-terminals: {ABC-CFG Note-Rest accidental octave Note duration broken mult slash div}

Start symbol: ABC-CFG

Productions:

1) ABC-CFG → Note-Rest
2) Note-Rest → Z duration
3) Note-Rest → Z
4) Note-Rest → z duration
5) Note-Rest → z
6) Note-Rest → X duration
7) Note-Rest → X
8) Note-Rest → Note
9) Note-Rest → Note duration
10) Note-Rest → Note octave duration
11) Note-Rest → Note accidental octave duration
12) Note-Rest → Note octave
13) Note-Rest → Note accidental octave
14) Note-Rest → Note broken Note
15) Note → A | B | C | D | E | F | G | a | b | c | d | e | f | g
16) duration → mult | slash | mult / div
17) slash → /
18) slash → / slash
19) mult → NZD
20) mult → NZD DIGIT
21) div → NZD
22) div → NZD DIGIT
23) DIGIT → 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
24) octave → , | , octave | ‘ | ‘ octave
25) accidental → ^ | ^ ^ | = | _ | _ _
26) broken → > | > broken | < | < broken
27) NZD → 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

Derivations:

1. Derive D, from the start symbol.

1) ABC-CFG => Note-Rest (1)
2) => Note octave (12)
3) => D octave (15d)
4) => D , (24a)

2. Derive b^’’ from the start symbol.

1) ABC-CFG => Note-Rest (1)
2) => Note accidental octave (13)
3) => b accidental octave (15i)

```
4) => b ^ octave                          (25a)
5) => b ^ ' octave                        (24d)
6) => b ^ ' '                             (24c)
```

3. Derive D,,', from the start symbol.

```
1) ABC-CFG => Note-Rest                   (1)
2) => Note octave                         (12)
3) => D octave                            (15d)
4) => D , octave                          (24b)
5) => D , , octave                        (24b)
6) => D , , ' octave                      (24d)
7) => D , , ' ,                           (24a)
```

4. Derive C5/12 from the start symbol.

```
1) ABC-CFG => Note-Rest                   (1)
2) => Note duration                       (9)
3) => C duration                          (15c)
4) => C mult / div                        (16c)
5) => C NZD / div                         (19)
6) => C 5 / div                           (27e)
7) => C 5 / NZDIGIT DIGIT                 (22)
8) => C 5 / 1 DIGIT                       (27a)
9) => C 5 / 1 2                           (23c)
```

5. Derive e ''9 from the start symbol.

```
1) ABC-CFG => Note-Rest                   (1)
2) => Note accidental octave duration     (11)
3) => e accidental octave duration        (15l)
4) => e _ octave duration                 (25d)
5) => e _ ' octave duration               (24d)
6) => e _ ' ' duration                    (24c)
7) => e _ ' ' mult                        (16a)
8) => e _ ' ' NZD                         (19)
9) => e _ ' ' 9                           (27i)
```

6. Derive A / / from the start symbol.

```
1) ABC-CFG => Note-Rest                   (1)
2) => Note duration                       (9)
3) => A duration                          (15a)
4) => A slash                             (16b)
5) => A / slash                           (18)
6) => A / /                               (17)
```

7. Derive c < d from the start symbol.

```
1) ABC-CFG => Note-Rest                   (1)
2) => Note broken Note                    (14)
3) => c broken Note                       (15j)
4) => c < Note                            (26c)
5) => c < d                               (15k)
```

8. Derive Z4 from the start symbol.

```
1) ABC-CFG => Note-Rest                   (1)
2) => Z duration                          (2)
```

3) => Z mult (16a)
4) => Z NZD (19)
5) => Z 4 (27d)

9. Derive z1/4 from the start symbol.

1) ABC-CFG => Note-Rest (1)
2) => z duration (2)
3) => z mult / div (16c)
4) => z NZD / div (19)
5) => z 1 / div (27a)
6) => z 1 / NZD (21)
7) => z 1 / 4 (27d)

CFG for JFugue Notation Notes and Rests:

Terminals: {A B C D E F G R 0 1 2 3 4 5 6 7 8 9 10 [ ] w h q i s t x o w . b # /}

Non-terminals: {JFugue Note-Rest note octave duration MIDI DIGIT NZD MTD D3 D2 D1 RAT-NUM DEC-NUM NUM accidental nDuration lDuration}

Start symbol: JFugue

Productions:

1) JFugue → Note-Rest
2) Note-Rest → note
3) Note-Rest → note addon
4) addon → duration | accidental | octave | duration addon | accidental addon | octave addon
5) Note-Rest → R duration
6) Note-Rest → R
7) accidental → # | # # | b | b b
8) octave → DIGIT | 10
9) note → C | D | E | F | G | A | B | [MIDI]
10) MIDI → D1 | D2 | D3
11) D1 → DIGIT
12) D2 → NZDIGIT DIGIT
13) D3 → 1 0 DIGIT
14) D3 → 1 1 DIGIT
15) D3 → 1 2 LTD-DIGIT
16) DIGIT → 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
17) NZD → 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
18) MTD → 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7
19) duration → lDuration | nDuration
20) nDuration → / RAT-NUM
21) RAT-NUM → NUM . DEC-NUM
22) NUM → DIGIT
23) NUM → DIGIT NUM
24) DEC-NUM → NUM
25) lduration → lduration lduration
26) lduration → w | h | q | i | s | t | x | o | w. | h. | q. | i. | s. | t. | x. | o.

Derivations:

1. Derive C from the start symbol.

```
1) JFugue => Note-Rest                    (1)
2) => note                                (2)
3) => C                                   (9a)
```

2. Derive Aw from the start symbol.

```
1) JFugue => Note-Rest                    (1)
2) => note addon                          (3)
3) => A addon                             (9f)
4) => A duration                          (4a)
5) => A lDuration                         (19a)
6) => A w                                 (26a)
```

3. Derive E7h from the start symbol.

```
1) JFugue => Note-Rest                    (1)
2) => note addon                          (3)
3) => E addon                             (9c)
4) => E octave addon                      (4c)
5) => E DIGIT addon                       (8a)
6) => E 7 addon                           (16h)
7) => E 7 duration                        (4a)
8) => E 7 lduration                       (19a)
9) => E 7 h                               (26b)
```

4. Derive [60]wq from the start symbol.

```
1) JFugue => Note-Rest                    (1)
2) => note addon                          (3)
3) => [ MIDI ] addon                      ()
4) => [ D2 ] addon                        ()
5) => [ NZD DIGIT ] addon                 ()
6) => [ 6 DIGIT ] addon                   ()
7) => [ 6 0 ] addon                       ()
8) => [ 6 0 ] duration                    ()
8) => [ 6 0 ] lduration                   ()
9) => [ 6 0 ] lduration lduration         ()
10) => [ 6 0 ] w lduration                ()
11) => [ 6 0 ] w q                        ()
```

5. Derive G8i. from the start symbol.

```
1) JFugue => Note-Rest                    (1)
2) => note addon                          (3)
3) => G addon                             ()
4) => G octave addon                      ()
5) => G DIGIT addon                       ()
6) => G 8 addon                           ()
7) => G 8 duration                        ()
8) => G 8 lduration                       ()
9) => G 8 i .                             ()
```

6. Derive Bb6/3.0 from the start symbol.

```
1) JFugue => Note-Rest                          (1)
2) => note addon                                ()
3) => B addon                                   ()
4) => B accidental addon                        ()
6) => B b addon                                 ()
7) => B b octave addon                          ()
8) => B b DIGIT addon                           ()
9) => B b 6 duration                            ()
10) => B b 6 nDuration                          ()
11) => B b 6 / RAT-NUM                          ()
12) => B b 6 / NUM . DEC-NUM                    ()
13) => B b 6 / DIGIT                            ()
14) => B b 6 / 3 . DEC-NUM                      ()
15) => B b 6 / 3 . 0                            ()
```

7. Derive Bb6www from the start symbol.

```
1) JFugue => Note-Rest                          (1)
2) => note addon                                ()
3) => B addon                                   ()
4) => B accidental addon                        ()
5) => B b addon                                 ()
6) => B b octave addon                          ()
7) => B b DIGIT addon                           ()
8) => B b 6 addon                               ()
9) => B b 6 duration                            ()
10) => B b 6 lduration                          ()
11) => B b 6 lduration lduration                ()
12) => B b 6 w lduration                        ()
13) => B b 6 w lduration lduration              ()
14) => B b 6 w w lduration                      ()
15) => B b 6 w w w                              ()
```

8. Derive R from the start symbol.

```
1) JFugue => Note-Rest                  (1)
2) => R                                 ()
```

9. Derive Rq from the start symbol.

```
1) JFugue => Note-Rest                  (1)
2) => R duration                        ()
3) => R lduration                       ()
4) => R q                               ()
```