

```

module Move where

import ParseLib

-----

data Move = M Piece (Maybe Disambiguation) (Maybe Capture) Square (Maybe
Promotion) (Maybe Check)
    | CastlingLeft
    | CastlingRight deriving (Eq, Show)

data Piece = N | B | R | Q | K | P deriving (Eq, Show)

data Disambiguation = File File | Rank Rank | Square Square deriving (Eq,
Show)

data Capture = Capture deriving (Eq, Show)

data File = Fa | Fb | Fc | Fd | Fe | Ff | Fg | Fh deriving (Eq, Show)

data Rank = R1 | R2 | R3 | R4 | R5 | R6 | R7 | R8 deriving (Eq, Show)

type Square = (File, Rank)

type Promotion = Piece

data Check = Check | Mate deriving (Eq, Show)

-----

parseMove  :: Parser Char Move
parseMove  =  M
            <$> parsePiece
            <*> parseDisambiguation
            <*> parseCapture
            <*> parseSquare
            <*> parsePromotion
            <*> parseCheck
            <|> const CastlingLeft
            <$> token "0-0-0"
            <|> const CastlingRight
            <$> token "0-0"

parsePiece  :: Parser Char Piece
parsePiece  =  const N <$> symbol 'N'
            <|> const B <$> symbol 'B'
            <|> const R <$> symbol 'R'
            <|> const Q <$> symbol 'Q'
            <|> const K <$> symbol 'K'
            <|> succeed P

parseDisambiguation  :: Parser Char (Maybe Disambiguation)
parseDisambiguation  =  option
            ( (Just . File) <$> parseFile
            <|> (Just . Rank) <$> parseRank
            <|> (Just . Square) <$> parseSquare
            )

```

Nothing

```
parseCapture :: Parser Char (Maybe Capture)
parseCapture = option (const (Just Capture) <$> symbol 'x')
                Nothing
```

```
parseSquare :: Parser Char Square
parseSquare = (\f r -> (f,r)) <$> parseFile <*> parseRank
```

```
parsePromotion :: Parser Char (Maybe Promotion)
parsePromotion = option ((\s p -> Just p) <$> symbol '=' <*> parsePiece)
                    Nothing
```

```
parseCheck :: Parser Char (Maybe Check)
parseCheck = option ( const (Just Check) <$> symbol '+'
                    <|> const (Just Mate) <$> symbol '#'
                    )
                Nothing
```

```
parseFile :: Parser Char File
parseFile = const Fa <$> symbol 'a'
          <|> const Fb <$> symbol 'b'
          <|> const Fc <$> symbol 'c'
          <|> const Fd <$> symbol 'd'
          <|> const Fe <$> symbol 'e'
          <|> const Ff <$> symbol 'f'
          <|> const Fg <$> symbol 'g'
          <|> const Fh <$> symbol 'h'
```

```
parseRank :: Parser Char Rank
parseRank = const R1 <$> symbol '1'
          <|> const R2 <$> symbol '2'
          <|> const R3 <$> symbol '3'
          <|> const R4 <$> symbol '4'
          <|> const R5 <$> symbol '5'
          <|> const R6 <$> symbol '6'
          <|> const R7 <$> symbol '7'
          <|> const R8 <$> symbol '8'
```

```
test = parseMove "Nexf7"
```