Parser Brief

COMP0009: Logic and Database Theory

October 20, 2023

1 Technical Brief

In this course, you will be tasked with writing a propositional and first order tableau, using Python. To assist you with this task, we have decided to release the language specification that we will be using for the coursework early, so you can have some time to write the parser before actually implementing the tableau program. We will also be releasing a parser verification program, which we will be creating shortly, where you will be able to submit your parsers, to test that they indeed parse the grammar as intended. Once the verification program is complete, you will be instructed on how to modify your parsing routine to have it be testable with our online testing tool.

IMPORTANT: Your file must not include **any** import statements.

Below we define both the language of propositional and first order logic. We limit our propositional letters to p, q, r, s, and our binary connectives to conjunctions, disjunctions, and implications. White space or extra brackets are not allowed in formulas.

```
FMLA := PROP                       (Proposition)
      | ~FMLA                      (Negation)
      | (FMLA*FMLA)                (Binary Connective)

PROP := p|q|r|s    * := /\|\/|=>   (and, or, implies)
```

Similarly for FOL, we limit our variables to x, y, z, w, no function symbols, and the only predicates are binary predicates P, Q, R, S.

```
var := x|y|z|w
PRED := P|Q|R|S      * := /\|\/|=>
FMLA := PRED(var,var) (Atom)
      | ~FMLA         (Negation)
      | EvarFMLA      (Existentially Quantified)
      | AvarFMLA      (Universally Quantified)
      | (FMLA*FMLA)   (Binary Connective)
```

If you have any questions about the coursework, please don't hesitate to ask on moodle or email me directly at ylbuzoku@cs.ucl.ac.uk