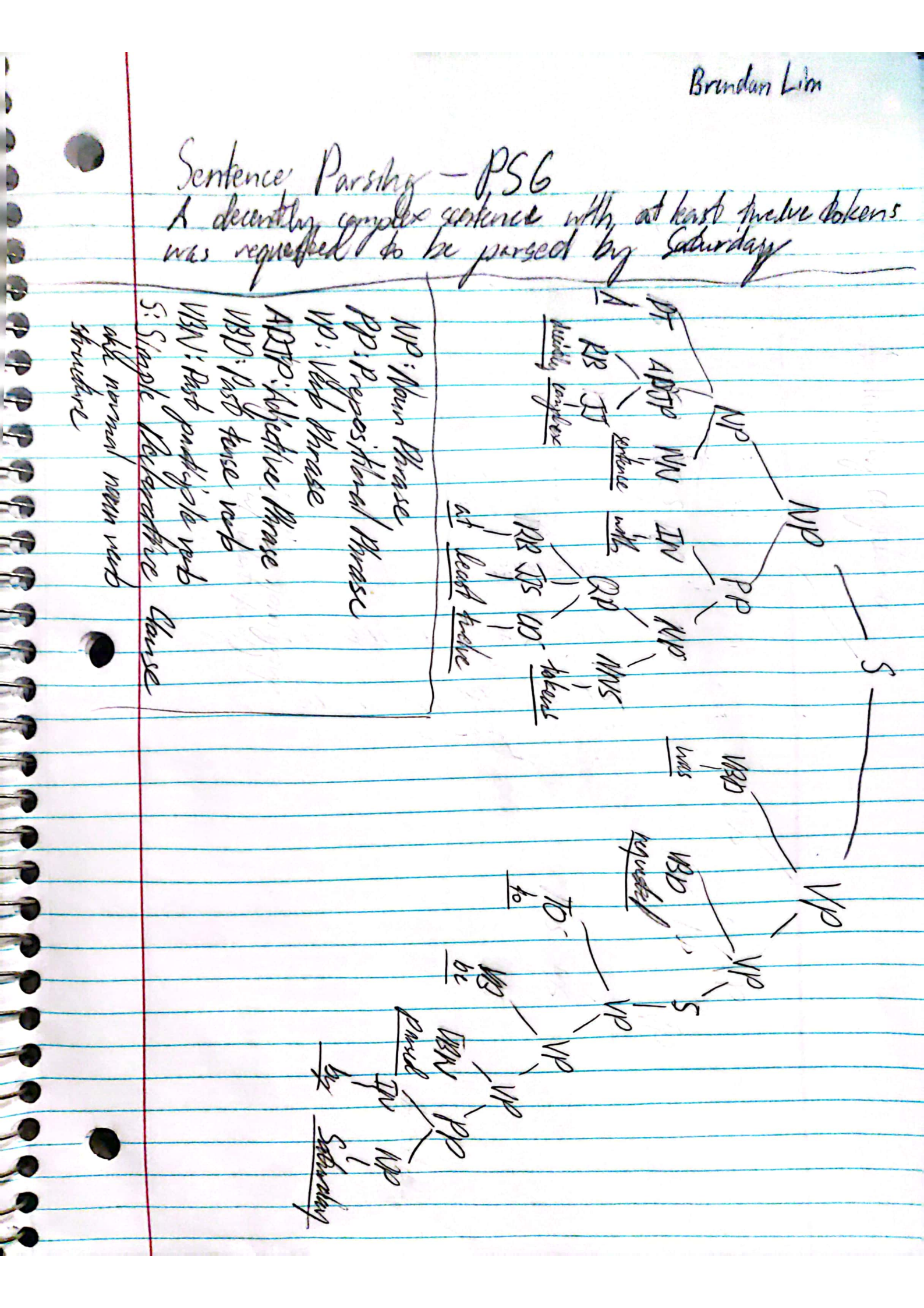
Brandan Lin

# Sentence Parsing – PSG

A decently complex sentence with at least twelve tokens was requested to be parsed by Saturday



# Sentence Parsing - Dependency Parse

det: Relation between determiner and its noun
advmod: non-clausal adverb/adverb headed phrase that serves to modify a word
amod: Adjective phrase that modifies the NP

nsubj:pass: Nominal subject in the passive
nmod: Noun which modifies another noun
obl:npmod: When a noun-phrase is used as an adverbial modifier
case: Case marking elements

aux:pass: passive auxiliary of a main word

mark: Introduces a finite subordinate clause to another
xcomp: A predicative/clausal complement of a verb or adjective with no subject
obl: Nominal noun functioning as a verb, adjective, adverb

# Sentence Parsing - SRL Parse

A decently complex sentence with at least twelve tokens was requested to be parsed by Saturday

'was' and 'be': Auxiliary verbs, no arguments

requested:

- 'A decently complex sentence with at least twelve tokens': Arg1
- 'to be parsed by Saturday': ArgM-PRD

parsed:

- 'A decently complex sentence with at least twelve tokens': Arg1
- 'by Saturday': ArgM-TMP

Arg1: Passive Actor

ArgM-PRD: Secondary predication modifier

ArgM-TMP: Temporal modifier

# Sentence Parsing - Pros and Cons

PSG trees are really good at breaking down sentences and determining parts of speech, however they don't go any deeper, which makes it hard to glean anything more from the sentence.

The dependency graph is my favorite because it shows relations between words and inside phrases. But it's hard to read because it jumps back and forth and words can have multiple dependencies

The SRL parsing was cool because it goes beyond breaking down sentence structure, but that's also its downfall. The SRL parser I used also detected auxiliary verbs by mistake.