

Nikola Peevski

Homework 3

CS536

This homework is about the language of *set expressions* defined as follows:

1. A a hash-separated list of zero or more countries enclosed in curly braces is a set expression.

Valid examples:

- o {}
- o {Australia}
- o {England # Australia}
- o ({England # Asia} \cap {Australia}) \cup {Antarctica}

2. If S1 and S2 are both set expressions, then so are each of the following:

S1 \cup S2

S1 \cap S2

S1 + S2

(S1)

Question1.

Write an unambiguous CFG for the language of set expressions so that parse trees correctly reflect the precedences and associativities of the operators. The terminals used are already for tokens. Use lower-case names for nonterminals and use the following terminals:

```
COUNTRY // one country in a set
U // union
 $\cap$  // intersection
+
(
)
{
}
#
// left paren
// right paren
// left curly brace
// right curly brace
// hash
```

sExpr -> sTerm PLUS sTerm | sTerm

sTerm -> sFinal UNION sFinal| sFinal INTERSECTION sFinal| sFinal

sFinal -> LPAR sExp RPAR | LPAR sSet RPAR | | sSet

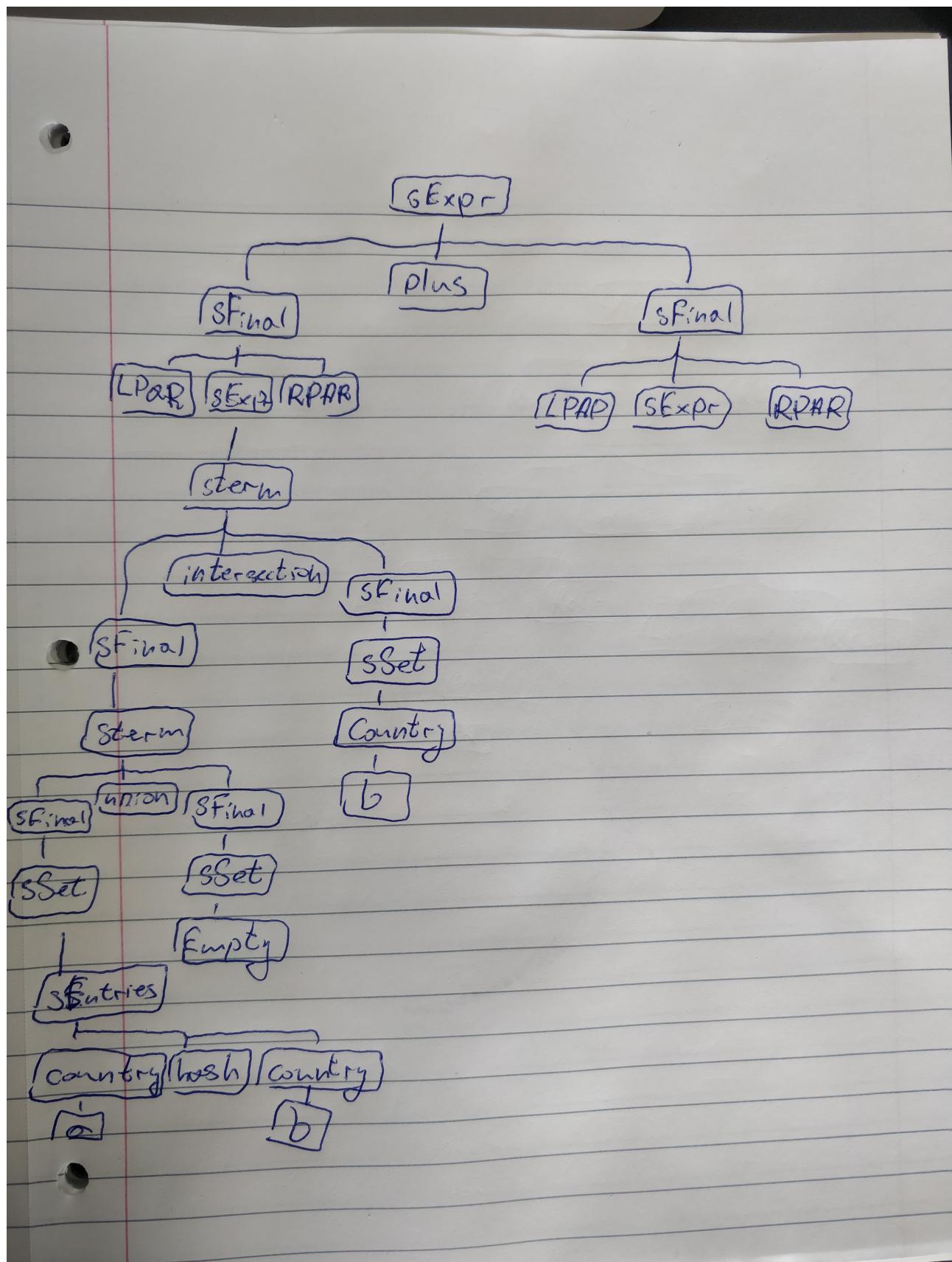
sSet -> LCRUL sEntries RCURL

sEntries -> country HASH country | country | EMPTY

Question2.

Draw a parse tree for the string. You could use COUNTRY(country) to represent a COUNTRY terminal with its actual name. For example, use COUNTRY(a) for a terminal name "a".

({ a # b } \cup { } \cap { b }) + ({ a } \cap { b })



Nikola Peevski

Homework 3

CS536

Due to lack of space I have omitted the LCURL and RCURL on sEntries in the bottom left and have omitted sEntries from sFinal -> sSet -> Empty/ sFinal-> sSet-> Country