Course 2BA1: Hilary Term 2003
Section 5: Formal Languages

David R. Wilkins

Contents

f0; 1g of binary digits, or the set f0; 1; 2; 3; 4; 5; 6; 7; 8; 9g of decimal digits.)

For any natural number *n*, we de ne a *word* of lend5F15 5,9.539.955 Tf 127.528 0 Td[(n)

Note that $|w_1 \quad w_2| = |w_1| + |w_2|$ for all words w_1 and w_2 over some alphabet.

The operation   of concatenation on the set of words over some alphabet is not commutative if that alphabet has more than one element. Indeed if a and b are distinct elements of this alphabet then $a \quad b$ is the string ab, and $b \quad a$ is the string ba, and therefore $a \quad b \neq b \quad a$.

Let w_1, w_2 and w_3

*h*T*i* represents the de nite article `the' (which will subsequently replace it);

*h*N*i* represents a noun chosen from the set the set *f*dog*;* cat*g*;

*h*Adj*i* represents an adjective chosen from the set *f*black*;* old*;* small*g*;

*h*V*i* represents a verb chosen from the set *f*saw*;* chased*g*;

Each of these productions speci es that the entity on the left hand side of the arrow may be replaced by the string on the right hand side of the arrow.

We may apply these productions successively, of h

*h*V*i* represents the verb chosen from the set *f* are applied, one at a time, to transform strings made up of terminals and

Let p and q be Boolean variables. The *conjunction* p ^ q of p and q is true if and only if both p and q are true. The *disjunction* p _ q of p and q

p	*q*	*r*	*p* ^ *q*	*q* _ *r*	(*p* ^ *q*) _ *r*	*p* ^ (*q* _ *r*)

determine in the usual fashion the order in which the binary operations are to be performed and the subformulae to which they are to be applied. We could introduce characters T and F to denote the Boolean constants `true' and `false' respectively. It remains to consider how Boolean variables are to be represented. We could certainly use single letters *p*, *q*, *r*, *s*. But this would only enable us to write down formulae with at most four distinct propositional variables. Were we to use single letters from the English alphabet in both upper and lower case to denote propositional variables, this would restrict us to formulae with at most fty-two distinct Boolean variables. But there should be no limit to the number of distinct Boolean variables that we could introduce into a well-formed formula. We therefore need a scheme for representing unlimited quantities of Boolean variables within our formula. W[illegible]ould do 32(therr)-g1(w)6,

Our grammar now has productions to generate any atomic formula. We ambiguity in the cases when G is atomic, the negation of a well-formed formula or a conjunction of well-formed formulae. (Our rules of precedence ensure that $:G$

We can then apply further productions in order to obtain formulae such as $p \wedge \neg q$ and $p' \wedge p'' \wedge p'''$.

Finally we have to specify the productions for handling disjunction. These are analogous to those for conjunctions, and are the following: —

$$\begin{array}{rcl}
\langle\text{disjunction}\rangle & \rightarrow & \langle\text{df}\rangle \vee \langle\text{df}\rangle \\
\langle\text{df}\rangle & \rightarrow & \langle\text{atom}\rangle \\
\langle\text{df}\rangle & \rightarrow & \langle\text{negation}\rangle \\
\langle\text{df}\rangle & \rightarrow & \langle\text{disjunction}\rangle \\
\langle\text{df}\rangle & \rightarrow & (\langle\text{w}\quad\rangle
\end{array}$$

*h*conjunction*i* *!* *h*

$$\begin{aligned}
\langle\text{variable}\rangle &\to \langle\text{letter}\rangle\\
\langle\text{letter}\rangle &\to p\\
\langle\text{letter}\rangle &\to q\\
\langle\text{letter}\rangle &\to r\\
\langle\text{letter}\rangle &\to s
\end{aligned}$$

The well-formed formulae of the Propositional Calculus are those words F over the alphabet

) (p ^ r^{0}) _ (hconjunctioni)
) (p ^ r^{0}) _ (hcfi ^ hcfi)
) (p ^ r^{0}) _ ((hw i) ^ hcfi)
) (p ^ r^{0}) _ ((hcompoundi) ^ hcfi)
) (p ^ r^{0}) _ ((hnegationi) ^ hcfi)
) (p ^ r^{0}) _ ((: hnfi) ^ hcfi)
) (p ^ r^{0}) _ ((: hatomi) ^ hcfi)
) (p ^ r^{0}) _ ((: hvariablei) ^ hcfi)
) (p ^ r^{0}) _ ((: hvariablei^{0}) ^ hcfi)
) (p ^ r^{0}) _ ((: hvariablei^{00}) ^ hcfi)
) (p ^ r^{0}) _ ((: hletteri^{00}) ^ hcfi)
) (p ^ r^{0}) _ ((: r^{00}) ^ hcfi)
) (p ^ r^{0}) _ ((: r^{00}) ^ hconjunctioni)
) (p ^ r^{0}) _ ((: r^{00}) ^ hcfi ^ hcfi)
) (p ^ r^{0}) _ ((: r^{00}) ^ hatomi ^ hcfi)
) (p ^ r^{0}) _ ((: r^{00}) ^ hvariablei ^ hcfi)
) (p ^ r^{0}) _ ((: r^{00}) ^ hletteri ^ hcfi)
) (p ^ r^{0}) _ ((: r^{00}) ^ q ^ hcfi)
) (p ^ r^{0}) _ ((: r^{00}) ^ q ^ hnegationi)
) (p ^ r^{0}) _ ((: r^{00}) ^ q ^ : hnfi)
) (p ^ r^{0}) _ ((: r^{00}) ^ q ^ : hatomi)
) (p ^ r^{0}) _ ((: r^{00}) ^ q ^ : hvariablei)
) (p ^ r^{0}) _ ((: r^{00}) ^ q ^ : hvariablei^{0})
) (p ^ r^{0}) _ ((: r^{00}) ^ q ^ : hvariablei^{00})
) (p ^ r^{0}) _ ((: r^{00}) ^ q ^ : hvariablei^{000})
) (p ^ r^{0}) _ ((: r^{00}) ^ q ^ : hletteri^{000})
) (p ^ r^{0}) _ ((: r^{00}) ^ q ^ : r^{000})

Remark i The grammar we have constructed to describe the well-formed for-parentheses, and that no subformula is enclosed by itself within two or more sets of parentheses. This modi ed grammar is expressed in Backus-Naur form as follows: |

*h*w *i*	*!*	*h*atom*i j h*compound*i*
*h*atom*i*	*!*	*h*constant*i j h*variable*i*
*h*compound*i*	*!*	*h*negation*i j h*conjunction*i j h*disjunction*i*
*h*negation*i*	*!*	*: h*nf*i*
*h*nf*i*	*!*	*h*atom*i j h*negation*i j* (*h*compound*i*)
*h*conjunction*i*	*!*	*h*cf*i* ^ *h*cf*i*
*h*cf*i*	*!*	*h*atom*i j h*negation*i j h*conjunction*i j* (*h*compound*i*)
*h*disjunction*i*	*!*	*h*df*i* _ *h*df*i*
*h*df*i*	*!*	*h*atom*i j h*negation*i j h*disjunction*i j* (*h*compound*i*)
*h*constant*i*	*!*	T *j* F
*h*variable*i*	*!*	*h*letter*i j h*variable*i*^{0}
*h*letter*i*	*!*	*p j q j r j s*

5.4 Context-Free Grammars

We have discussed examples of context-free grammars. We now present and discuss a formac de nition of such grammars.

De nition A *context-free grammar* (*V; A; h*S*i; P*) consists of a nite set *V*, a subset *A* of *V*, an element *h*S*i* of *V nA*, and a nite subset *P V nA*) *V* .

Let (*V; A; h*S*i; P*) be a context-free grammar. The elements of *A* are referred to as *terminals*. Let *N* = *V n A*. The elements of *N* are referred to as *nonterminals*. The nonterminal *h*S*i* is the *start symbol*. The set *N* of nonterminals is non-empty since *h*S*i 2 N*.

The nite set *P* speci es the *productions* production $\langle T \rangle \rightarrow w$ of the grammar such that w

As in the case of context-free grammars, the elements of A are referred to as *terminals*, the elements of $V \ n A$ are referred to as *nonterminals*, the nonterminal hSi is the *start symbol* and the elements of P specify the *productions* of the grammar. The pro speci ed bp a1 element ($r; w$) of P is denoted bp $r \ ! \ w$.wever the left hand side r of a pro $r \ ! \ w$ in a phrase strF15ure grammar need not consist solely of a single nonterminal, but map be a nite string r of elements of V , provided that this string r contains at least one nonterminal. (Note that V denotes the set of all nite words over the alphabet V whose elements are terminals and nonterminals, A denotes the set of all nite words consisting entirely of terminals, and thus $V \ 1A$ denotes the set of all nite words belonging to V which contain atterminal.) Det w^0 tand 11 w^{00} $r5 \ w$ wthe

an.955 Tf 0 0 Td[(As)-374(in)-96.8(F37 7.97 Tf 9.47 -4.389 Tfer414ni14.446 551 denote1

Example Let *L*

where hAi and hBi represent nonterminals, b represents a terminal, and $"$ denotes the empty word. A regular grammar is said to be in *normal form* if

$$\begin{array}{lcl} hBi & ! & 2hBi \\ & \vdots & \\ hBi & ! & 9hBi \\ hBi & ! & '' \end{array}$$

De nition Let $(S; A; i; t; F)$ be a nite state acceptor. A language L over the alphabet A is said to be *recognized* or *accepted* by the nite state acceptor if L