```perl
#!/usr/local/bin/perl

if ($#ARGV == -1) {
    $sent = "the woman saw a little boy with a telescope";
#    $sent = "I put the can in the water";
    $grammar = "chart.grm";
} elsif ($#ARGV == 0) {
    $sent = $ARGV[0];
    $grammar = "eng.grm";
} elsif ($#ARGV == 1) {
    $sent = $ARGV[0];
    $grammar = $ARGV[1];
}


# Load in Grammar
@rules = ();

# Add seed rule

# Load in rules, refer to them later by index
$rulenum = 0;
open(GRM, $grammar) or die "Can't open grammar file $grammar\n";
while (<GRM>) {
    next if m/^\s*$/;
    if (m/Terminals/g) {
        while ($line = <GRM>) {
          if ($line =~ m/rules/i) { last; }
          next if $line =~ m/^\s*$/;
          chomp($line);
          $terminals{lc($line)} = 1;
        }
        while (<GRM>) {
            chomp;
            next if m/^\#/;
            my(@fields) = split;
            if (scalar(@fields) == 2 and
                $terminals{lc($fields[1])}) {
                $wordpos{$fields[1]} .= "$fields[0] ";
            } else {
                $rules[$rulenum] = \@fields;
                $rulenum++;
            }
            $nonterms{$fields[0]} = 1;
        }
    }
}
close(GRM);

# Create an index for all non-terminals
$j = 0;
foreach $nonterm (sort keys %nonterms) {
    $nonterms[$j] = $nonterm;
    $nonterms{$nonterm} = $j;
    $j++;
}

foreach $wordps (keys %wordpos) {
    $wordpos{$wordps} =~ s/\s$//;
#    print "$wordps ", $wordpos{$wordps}, "\n";
}



#foreach $rule (@rules) {
#  print @{$rule}, "\n";
#}

print $sent, "\n";
@words = split(/\s+/, $sent);
$sentlen = scalar @words;

# Initialize agenda and active arcs
@agenda = ();
@activearcs = ();
@chart = ();

# Agenda, key, chart form: Constituent index, start position, end position
# Active arc form : Rule index, start position, end position, dot position
$i = 0;
$accepted = 0;
while ($i <= $sentlen) {
    # If agenda is empty, increment counter and add all POS for word at i to agen
    print "\nI is ", $i+1, "\n";
    if (scalar(@agenda) == 0 and $i < $sentlen) {
        $i++;
        my(@poses) = split(/\s/, $wordpos{$words[$i-1]});
        foreach $pos (@poses) {
            $newkey = sprintf("%d,%d,%d", $nonterms{$pos}, $i, $i+1);
            push(@agenda, $newkey);
            print "Adding into agenda: $pos,$i,", $i+1, "\n";
        }
    } else {
        last;
    }

    while (scalar(@agenda) > 0 and !$accepted) {
        # Pick a key constituent from the agenda
        $key = pop(@agenda);
        my($keyconst, $keystart, $keyend) = split(/,/, $key);
        print "Using key ", $nonterms[$keyconst], ",$keystart,$keyend\n";

        # Add each grammar rule starting with key to active arcs
        for ($k = 0; $k <= $#rules; $k++) {
            if ($rules[$k]->[1] eq $nonterms[$keyconst]) {
                push @activearcs, "$k,$keystart,$keystart,1";
                print " Adding active arc " . join(" ", @{$rules[$k]}) . " $keyst
            }
        }

        # Look through all the active arcs, looking for ones to extend with key
        # and adding completed ones into agenda
        foreach $activearc (@activearcs) {
            my($ruleind, $rulestart, $ruleend, $ruledot) = split(/,/, $activearc)
            next if ($nonterms[$keyconst] eq "");
            if ($rules[$ruleind]->[$ruledot] eq $nonterms[$keyconst] and
                $ruleend == $keystart) {
                $ruledot++;
                $ruleend = $keyend;
                print " Extending ", join(" ", @{$rules[$ruleind]}), " $rulestart
                if ($ruledot == scalar(@{$rules[$ruleind]})) {
                    # Add LHS of completed active arc into agenda
                    my($newkey) = $nonterms{$rules[$ruleind]->[0]} . ",$rulestart
                    my($addkey) = 1;
                    for ($m = 0; $m <= $#agenda; $m++) {
                        if ($agenda[$m] eq $newkey) {
                            $addkey = 0;
                        }
                    }
                    if ($addkey) {
                        push @agenda, $newkey;
                        print " Adding LHS ", $nonterms[$nonterms{$rules[$ruleind
```

```
                    }
                } else {
                    my($newarc) = "$ruleind,$rulestart,$ruleend,$ruledot";
                    my($addarc) = 1;
                    for ($m = 0; $m <= $#activearcs; $m++) {
                        if ($activearcs[$m] eq $newarc) {
                            $addarc = 0;
                        }
                    }
                    if ($addarc == 1) {
                        #print "Adding arc $newarc\n";
                        push @activearcs, $newarc;
                    }
                }

            }

        }


        # Add key into chart
        push @chart, $key;

        if ($key eq sprintf("%d,%d,%d", $nonterms{"S"},1,$sentlen+1)) {
            print "$sent is accepted\n";
            $accepted = 1;
        }
    }
}


# Check to see if sentence was accepted
if (!$accepted) {
    print "$sent is not accepted\n";
}
```