Problem

If we disallow "-rules in CFGs, we can simplify the *DK*-test. In the simplified test, we only need to check that each of *DK*'s accept states has a single rule. Prove that a CFG without "-rules passes the simplified *DK*-test iff it is a DCFG.

Step-by-step solution

Step 1 of 1

Given:

Consider a context free grammar *C* which does not have ε rule. Apply DK-test on *C* to check whether it is a deterministic CFG or not.

DK Test:

This test is used to check whether CFG is deterministic CFG or not. Consider a CFG C, create DK an associated DFA. Check whether CFG is deterministic by checking the accept state of DK. Every accept states must have:

• Only 1 completed rule.

• There should be no dotted rule that is mean dot should not be immediately after the terminal.

Proof:

Assume a contrary that C is not a deterministic CFG to show the failure of DK-test. Consider a valid string ahb which has *h* as unforced handle. It may possible that some other valid string ahb' has another handle $\hat{h} \neq h$. Therefore ahb' can be rewrite as $\hat{a}\hat{h}\hat{b}$ where b' is a terminal.

• When $ah = \hat{a}\hat{h}$, this lead to change the reduce rule because h and $\hat{h}$ both are two different handle. Hence, ah take the DK to state which has two completed rule which does not satisfy the DK-test.

• When $ah \neq \hat{a}\hat{h}$, the one extends the other. Assume that proper prefix of $\hat{a}\hat{h}$ is ah. There are same arguments with the interchanged string and in place of b' use b.

Assume that *w* is an accepting state in which DK enters on ah input. The state *w* must be an accepting states because h is the handle of ahb. As *w* is the accepting state, so the transition arrow must terminate at w also $\hat{a}\hat{h}$ take DK to accepting state by *w*.

The transition must label with b' because $b' \in \Sigma^+$ also *C* does not have null rule. Hence *w* have dotted rule that is, dot exist immediately after terminal symbol which does not satisfy the DK-test.

Conclusion:

The DK test fails; therefore contradiction occurs that C is not a deterministic CFG. Hence CFG is a DCFG.

Comment