9. Tétel

a) Formális nyelvek

Ábécék, szavak, formális nyelvek

- ABC: általában véges, nem üres halmaz

- jelek: ABC elemei

- (ABC fölötti) szó: ABC jeleiből alkotott kifejezés (kis görög betű)

- szó hossz: a szó jelenék a számával egyezik meg

- üres szó jele az epszilon

- formális nyelv: az ABC jeleiből alkotható, meghatározott szavak halmaza

- állhat véges és végtelen sok szóból és tartalmazhatja az üres szót

- megadhatjuk felsorolással, tulajdonságokkal vagy a szavakat alkotó szabály szöveges leírásával

Műveletek szavakkal és nyelvekkel

- nyelveknél: kontextus szorzás

- szavaknál:

- konkatenáció (+):asszociatív, nem kommutatív

- hatványozás: n-szeres konkatenáció

- szavak tükrözése

- résszavak

- bonyolultság: szóhossz

- ABC-knél: komplexus szorzás: A \* B (egyik jel egyik ABC, másik jel másik ABC-ből

- lezárt, pozitív lezárt

Szintaxis-leíró eszközök

Generatív grammatikák

- ellenőrzésre használjuk

- G(V,W,S,P) - formális négyes

- V: terminális jelek

- W: nemterminális jelek (V diszjunkt W)

- S: kezdőszimbólum (S eleme W)

- P: helyettesítési szabályok halmaza (Descartes féle szorzattal)

Chomsky-féle osztályozás

- G(V,W,S,P) ez a kiindulás

- Mondatszerkezetű nyelvek: 0-ás típusú->bal oldalon legalább egy nemterminális szimbólum

- Környezetfüggű nyelvek: 1-es típusú->számít a környezete (gyakorlatban a fordítóprogramok szemantikájának ellenőrzésében használják) + megengedi az S->epszilont

- Környezetfüggetlen nyelve: 2-es típusú-> bal oldalán kizárólag egy nemterminális jel(szintaxis leírására és a szintaktika elemzésére használják) + megengedi az S->epszilont

- Reguláris nyelvek: 3-as típusú-> A->a A->Ba (használata a lexikális elemzés) + nem engedi az S->epszilont

Levezetési fák, elemzési stratégiák

- Szemantikai elemzésre (tartalom):

- segítségével eldönthetjük, hogy az adott szó generálható-e az adott nyelvben

- balról jobbra vagy felülről lefelé (üres szót kihagyva) a levélelemeken a vizsgált szó jelenik meg, akkor generálható

- kanonikus, baloldali levezetés (mindig a legbaloldalabbi nemterminális jelet választjuk)

- top-down, szintaxisfa építés

- bottom-up, redukció

- visszalépéses

- Szintaxis elemzők (nyelvtan értelmezése az adott programozási nyelven)

A véges és a verem-automaták

- nincs output szalagjuk, sem író-olvasó fejük

- G(K,V,delta,q0,F) - formális ötös véges

- K: belső állapotok véges halmaza

- V: input ABC (terminális jelek)

- delta: állapotátmenet függvény

- q0: speciális belső állapot (kezdőállapot)

- F: befejező (elfogadó állapotok halmaza)

- q0->olvasó fej beolvas aktuális jelet az input szalagról->jel+belső állapot=új állapot->egyel jobbra lépés->ismétlés->megállás, ha az olvasó fej lelép a szalagról->(F?)

- speciális komponense van a verem, a veremhez tartozó író-olvasó fej nem mozoghat szabadon, mindig a legfelső cellába írhat (LIFO)

- G(K,V,W,delta,q0,z0,F) - formális hetes

- W: verembe írható és onnan kiolvasható terminális és nemtermináls jelek

- z0: speciális jel, a verem üres szimbólum (ez van inicializáláskor a veremben)

- konfiguráció

A Turing gépek és változataik ismertetése

- nincs output szalagjuk, az input szalagról nem csak olvasni, hanem írni is képes

- a szalag mindkét irányban végtelen

- megállása: a delta függvény parcialitása miatt

- A(K,V,W,delta,q0,B,F) - formális hetes

- W: output szalag (ugyanaz mint az input szalag (V))

- B: blank jel

- konfiguráció

- Egyirányban végtelen szalaggal rendelkező (ekvivalens a mindkét irányúval)

- Többszalagú Turing-gépek (ez is lehet ekvivalens)

- Lineárisan korlátolt (mindkét irányban véges) -> amit ezzel ismerünk fel, az környezetfüggő nyelv

A delta leképezés tulajdonságai, megadási módjai a különböző automaták esetén

- Az átmenetfüggvény megadása automatától függően determinisztikus vagy nem-determinisztikus lehet,

Az automaták és a grammatikák kapcsolata

- 0: Turing-gép

- 1: Lineárisan korlátolt automata

- 2: Veremautomaták

- 3: Véges automaták

b) A rendszerfejlesztés technológiája

Szoftver életciklus

- megszületéstől az átadásig

- sosincs kész (ciklikusan meg-megújul)

- lépéseit a módszertanok határozzák meg

- 9 lépés

Tervezés, implementáció, tesztelés iterációk

- Tervezés: logikai (ideális rendszer) és fizikai rendszertervekből épül fel

- A fizikai a logikai finomítása a fizikai megszorításokkal

- Maga a rendszerterv a programozó szemszögéből van

- Implementáció: újrafelhasználhatóság, keretrendszer, termék = forráskód + programozói dokumentáció

- Tesztelési iterációk: tesztelés előtt biztosan van hiba, tesztelés után a tesztelt részekben nincs hiba (tesztelési alapelvek)

- Ez az iteráció: szoftverben lévő hibák megtalálása, javítása -> üzembehelyezés -> NŐ a termék minősége és megbízhatósága

- fejlesztői teszt, átadás-átvételi teszt, regressziós teszt

Rendszerfejlesztés eszközei, feladatkövetés, verziókövetés

- feladatkövetés, verziókövetés, modellező eszközök (pl. képernyő tervek, UML diagramok(use-case), IDE)

- Feladatkövetés (Trello, github):

- elődje a Gantt-diagramm

- vizualizálja a feladatokat és folyamatokat

- Kanban (WIP) és SCRUM (Sprintek és iterációk)

- Verziókövetés (github):

- állományok tartalmi változásának követése

- korábbi állapot, eltérések vizsgálata, conflict kezelése

- kifejezések: munka könyvtár, hely, helyi tároló, távoli tároló, branch, check-out, repo, clone, commit, push

Tesztelési technikák, tesztelés szintjei

- Feketedobozos (black-box) specifikáció alapú

- Fehérdobozos (white-box) strukturális, forráskód alapú (kódsorok, elágazások, metódusok)

- Szürkedobozos (grey-box) a forráskódnak csak egy ismert része alapján készülnek a tesztesetek

- Tesztelés szintjei:

- komponensteszt: külön-külön teszteli a rendszer komponenseit (unit-teszt->metódusok, modulteszt)

- integrációs teszt: komponensek között együttműködést teszteli (de lehet rendszer integrációs is, amikor más rendszerek közötti kölcsönhatásokat nézzük)

- rendszerteszt: minden komponenst együtt (egész rendszer tesztje) (3 dokumentációnak megfelelősségét) (feketedobozos teszt elfogultság elkerülése érdekében más cég végzi)

- átadás-átvételi teszt: felhasználók tesztje a kész rendszeren (alfa, béta, felhasználói á., üzemeltetői á.)

- Regressziós teszt: összes unit-teszt futtatása egy változtatás után