+/- 7(chomsky???) 17

29 30 31 32 33 34 Deduktiivne jama

1.Hulkade spetsifitseerimine. Hulk on samalaadsete objektide järjestamata kogum, mida käsitletakse kui tervikut.Hulka kuuluvaid objekte nim. hulga elementideks. Elemendi kuulumist hulka tähistatakse $a\in A$. Hulkade esitamiseks kasut. kahte viisi: 1.Elementide loetlemine, nt.: $A = \{1,2,3\}$, 2.Predikaadi abil e. avaldisega kujul $H = \{z|P(z)\}$, nt.: $L = \{x|3x^2-4x+2 = 0\}$. Tühja hulga tähis on $\varnothing$. Lõpliku hulga elementide arvu nim. hulga võimsuseks ja tähistatakse: $|A|$. **Tehted hulkadega**. $A\cup B$ – hulkade ühend, $A\cap B$ – hulkade ühisosa, $A\backslash B$ e. $A-B$ – hulkade vahe, A' – hulga täiend, $A\times B$ – rist korrutis: $A\times B = \{(a,b)|a\in A \,\&\, b\in B\}$. Hulkade võrdlemine. $A=B$ – hulgad on võrdsed e. sisaldavad samu elemente, $A\subseteq B$ – A on B osahulk, $A\subset B$ – A on B pärisosahulk ($A\neq B$). Hulga A kõigi alamhulkade hulka tähistatakse $\wp(A)$ või 2^A. Kui $|A| = n$, siis $|2^A| = 2^n$**. Hulgateooria paradoksid.** Predikaadi abil hulga määramisel võib tekkida vastuolu, nii nagu näiteks "Russelli habemeajaja" paradoksi puhul: tähistagu predikaat P(X) tingimust, mis on tõene, kui argumendina antud hulk X pole iseenda element ja väär vastasel juhul. Hulga $Y = \{X|P(X)\}$ korral predikaadi P kontrollimine viib vastuoluni: 1. Juhul, kui oletame, et $Y\in Y$, siis P(Y) on väär ning hulga Y määratluse põhjal $Y\notin Y$; 2. Kui oletame, et $Y\in Y$, siis P(Y) on tõene ning seega $Y\in Y$. See on üks paljudest hulgateooria paradoksidest, mis viitab selle teooria ebatäiuslikkusele.

2.Relatsioonid. 1.Seos A ja B vahel on alamhulk $R\subseteq A\times B$. 2.Seos hulgal A on alamhulk $R\subseteq A\times A$. 3.Asjaolu, et $(a,b)\in R$ tähistatakse aRb. Seost $R^{-1} = \{(b,a)|aRb\}$ nim. seose R pöördrelatsiooniks. **Ekvivalentsi- ja järjestusseosed.** DF: Seost R hulgal A nim. ekvivalentsiseoseks siis, kui: 1.R on refleksiivne ($\forall a\in A$ korral aRa), 2.R on sümmeetriline ($a,b\in A$ korral $aRb \Rightarrow bRa$), 3.R on transitiivne ($a,b,c\in A$ korral $aRb \,\&\, bRc \Rightarrow aRc$). DF: Olgu R ekvivalentsiseos hulgal A. Elemendiga $a\in A$ ekvivalentsete elementide hulka $[a] = \{b|aRb\}$ nim. elemendi ekvivalentsiklassiks. TR: Olgu R ekvivalentsiseos hulgal A. $\forall\ a\in A$ ja $b\in A$ korral kehtib kas $[a] = [b]$ või $[a]\cap[b] = \varnothing$. DF: Olgu R seos hulgal A. Seose R k-s aste R^k on määratud järgmiste tingimustega: 1.aR^1b kehtib siis, kui aRb, 2.aR^kb (k>1) kehtib siis, kui $\exists c\in A$, nii et aRc & $cR^{k-1}b$. DF:Olgu R seos hulgal A. Seose R transitiivseks sulundiks nim. seost $R^+\subseteq A\times A$, nii et aR^+b kehtib siis, kui $\exists i\geq 1$, mille korral kehtib aR^ib. DF: Olgu R seos hulgal A. Seose R refleksiivseks transitiivseks sulundiks nim. sellist seost R^*, nii et: 1.$aR^*a\ \forall\ a\in A$, 2. aR^*b kehtib siis, kui aR^+b, 3.R^* ei sisalda rohkem elementide paare kui on määratud viimase kahe tingimusega. TR: Kui R^+ ja R^* on vastavalt seose R transitiivne ja refleksiivne-transitiivne sulund, siis: 1.Kui R' on suvaline transitiivne seos, selline et $R\subseteq R'$, siis $R^+\subseteq R'$, 2. Kui R' on suvaline refleksiivne-transitiivne seos, selline et $R\subseteq R'$, $R^*\subseteq R'$. DF: Seos R määrab hulgal A osalise järjestuse, kui: 1.Ron transitivne, 2.R on irrefleksiivne, st. $\forall a\in A$ korral $(a,a)\notin R$. DF: Seos $R\subseteq A\times A$ on refleksiivne osaline järjestus hulgal A, kui: 1.R on transitiivne, 2.R on refleksiivne, 3.$(aRb) \,\&\, (bRa) \Rightarrow (a=b)$. DF: Refleksiivne seos $R\subseteq A\times A$ määrab hulgal A lineaarse järjestuse, kui kehtib üks järgmistest tingimustest: aRb, bRa või a=b.

3.Kujutused. DF: Kujutuseks M hulgast A hulka B nim. seost $M\subseteq A\times B$, nii et $(a,b)\in M \,\&\, (a,c)\in M \Rightarrow b=c$. Def.-s esitatud kujutust tähistatakse $M{:}A\rightarrow B$. Asjaolu $(a,b)\in M$ tähistatakse ka M(a) = b. Kujutuse $M{:}A\rightarrow B$ muutmispiirkond $Dom(M) = \{a|a\in A, \exists b\in B(b=M(a))\}$ ja määramispiirkond $Ran(M) = \{b|b\in B, \exists a\in A(M(a)=b)\}$. DF: Kujutust $M{:}A\rightarrow B$ nim.: 1.osaliseks, kui $Dom(M)\subset A$, 2.täielikuks, kui Dom(M) = A, 3.pealekujutuseks e. sürjektsiooniks, kui M on täielik ja Ran(M) = B, 4.üks-üheseks e. injektsiooniks, kui $\forall\ a,a'\in A$ ning $\forall\ b\in B$ korral kehtib seos $(f(a)=b \,\&\, f(a')=b) \Rightarrow a=a'$, 5.bijektsiooniks, kui ta on sürjektsioon ja injektsioon. **Hulga võimsus.** DF: Hulgad A ja B on võrdvõimsad, kui $\exists$ bijektiivne kujutus $M{:}A\rightarrow B$. DF: Lõpmatu hulk A on loenduv, kui ta on võrdvõimas naturaalarvude hulgaga N. Loenduva hulga võimsust tähistatakse sümboliga ω: $|N| = \omega$.

4.Programmeerimiskeelte klassid. Prog.keel on tähistuste ja reeglite süsteem algoritmide esitamiseks arvutile. Prog.keel on inimese ja masina vaheline suhtlemiskeel. Selle järgi, kuivõrd keel on inimesele sobival kujul, võib prog.keeli liigitada kolme klassi: 1.masinkeeled, 2.algoritmilised e. kõrgtaseme keeled, 3.teadmiste esitamise e. spetsifiseerimiskeeled. Masinkeele näideteks on: 1.masinkood – konkreetse arvuti käskude jada kodeerituna 2-nd, 8-nd, 16-nd-süsteemi arvudena. 2.autokood – konkreetsele arvutile orienteeritud prog.keel. Enne täitmist transleeritakse autokoodis esitatav programm masinkoodi. Programmi teisendamist vastavateks protsessori käskudeks nim. assembleerimiseks. Autokoodi nim. ka assemblerkeeleks. Masinkeeltes kirjutatakse iga uue arvuti süsteemne tarkvara. Algoritmiliste keelte näiteid: Fortran, Cobol, Algol-60, Basic, PL/1, Pascal, Lisp, Ada, C. Algoritmilistes keeltes esitatakse aritmeetlised arvutused algebraliste arvutustena, kasutatakse spetsiaalseid lausekonstruktsioone, saab erinevate objektide omadusi esitada kasutades erinevaid andmetüüpe, saab kirjeldada sisendeid/väljundeid. Teadmiste esitamise keeled on ette nähtud ülesannete spetsifitseerimiseks, ülesande lahendusalgoritmi koostab arvuti lähtudes ülesande kirjeldusest. Keelte näiteid: PROLOG, CLU, UTOPIST. Tänapäeval on mitmed spetsifitseerimiskeeled visuaalsed. See tähendab, et ülesande püstitamiseks arvutile tuleb hiirega ekraanile joonistada ülesande tingimusi esitav joonis, ning arvuti teisendab selle ise tekstilisele kujule.

5.Programmeerimiskeelte formaalne spetsifitseerimine. Arvutil on võimalik lahendada vaid neid ülesandeid, mille lahendusalgoritm on esitatav matemaatilise formalismi abil. Kõik meetodid keelte formaliseerimiseks vaatlevad lahus süntaksi ja semantika kirjeldamist. Keele süntaks väljendab lausete sismist struktuuri sõltumata lause tähendusest, semantika esitab lause tähenduse. Vahe keele süntaksi ja semantika vahel pole siiski range. Keele

süntakilisis omadusi võib vaadelda semantilistena ja vastupidi. Keele semantika esitamine on palju keerulisem kui süntaksi määratlemine. Nii on ka keele süntaksi formaliseerimise meetodid hästi tuntud, samal ajal kui semantika jaoks sobiv efektiivne formalism puudub. Üks formalisme süntaksi esitamiseks on näiteks süntaksdiagrammid e. Wirthi skeemid. Süntaksdiagramm on graaf, mis esitab keele lause või selle osa süntaktilise struktuuri. Grammatiliselt korrektse lause konstrueerimiseks tuleb graafi läbida mööda kaari. Enamasti esitatakse lause semantika tekstina loomulikus keeles. Süntaksi esitamisel on populaarsed ka Bacus-Nauri valemid, mis esitavad reeglid korrektse keele lausete koostamiseks fraasidest. Bacus-Nauri valemid on samaväärsed Wirthi skeemidega, võimaldades esitada ainult süntaksi. **Transleerimisprotsessi osad.** Semantika tuleb defineerida vastava lause tõlkimisprotseduuri e. transleerimisalgoritmi kaudu. Kõiki arvutiprogramme võib käsitleda translaatoritena, mis tõlgivad programmi sisendandmed neile vastavateks väljundandmeteks. Olgu sisendandmete esitamise keel L_1 ja tulemuste keel L_2. Siis võib programmi tööd e. transleerimisprotsessi vaadelda kujutusena T_r:L1→L2. Kujutuse T_r arvutil realiseerimise skeem sisaldab kolm etappi: 1.süntaksanalüüs, 2.semantiline analüüs, 3.teksti genereerimine.
6.Keel kui matemaatiline objekt. Tähistame sümboliga Σ teatud tähestiku. Sümbolitega a,b,c,… märgime tähestiku Σ tähti ja sümbolitega x,y,z,… stringe. DF: String tähestikus Σ on määratud järgmiselt: 1.ϵ - tühi string, 2.ax on string, kuix on tähestiku Σ string ja sümbol $a\in\Sigma$, 3.y on tähestiku Σ string vaid siis, kui ta rahuldab tingimusi 1 ja 2. Tähestiku kõigi stringide hulka tähistatakse Σ^*. Stringi x pöördstringiks x^R nim. sümbolite jada, mis saadakse x kirjutamisel tagantpoolt ettepoole. Stringide $x\in\Sigma^*$ ja $y\in\Sigma^*$ järjestkirjutamise operatsiooni nim. konkatenatsioonioperatsiooniks. Stringide x ja y konkatenatsiooni tähist. xy. Selle omadused: 1.(xy)z = x(yz), 2.$\epsilon x = x\epsilon = x$. Kui on olemas string s = xyz, siis x,y,z on alamstringid, kus x on prefiks ja z on sufiks. DF: Keel on alamhulk $L\subseteq\Sigma^*$. DF: Keelte $L_1\subseteq\Sigma_1^*$ ja $L_2\subseteq\Sigma_2^*$ konkatenatsiooniks nim. keelt $L\subseteq(\Sigma_1\cup\Sigma_2)^*$, kus $L = \{xy|x\in L_1, y\in L_2\}$. DF: Keele L iteratsiooniks nim. hulka $L^* = \cup_{n\geq0}L^n$, kus: 1.$L^0 = \{\epsilon\}$, 2.$L^n = L^{n-1}L$, kui n>0. DF: Olgu Σ_1 ja Σ_2 tähestikud. Kujutust $h:\Sigma_1^*\to\Sigma_2^*$ nim. homomorfismiks, kui: 1.$h(\epsilon) = \epsilon$, 2.$h(ax) = h(a)h(x)\ \forall\ x\in\Sigma_1^*\ \&\ a\in\Sigma_1$ korral.
7.Fraasistruktuuri grammatikad. Transleerimisprotsessi automatiseerimiseks vajatakse formaalset aparatuuri keele ja tema lausete fraasistruktuuri esitamiseks. Matemaatilises mõttes on keel teatud tähestik $T = \{a_0,a_1,\ldots,a_n\}$ stringide alamhulk $L = \{x|x\in T^*, P(x)\} \subseteq T^*$. Lisaks keele tähestikule Σ, mida nim. terminaalide tähestikuks, kasutatakse metatähiseid fraaside tähistamiseks. Fraase tähistavaid metsümbolite hulka nim. mitteterminaalide tähestikuks. Terminaalide tähistamiseks kasut. ladina tähestiku väiketähti ja numbrimärke ning mitteterminaalide tähistamiseks ladina tähestiku suurtähti, kusjuures S on algsümbol. Eeldatakse, et $\Sigma\cap N = \emptyset$. Stringe tähestikus $V = \Sigma\cup N$ nim. lausevormideks. Teksti genereerimisel kasut. lausevormi järkjärgulist teisendamist, kuni saadakse string, mis ei sisalda enam mitteterminaal. Teisendusreeglid esitatakse paaridena $\alpha\to\beta$, kus α ja β tähistavad lausevorme. Avaldist $\alpha\to\beta$ nim. prod.iks. Prod.i $\alpha\to\beta$ olemasolu korral võib igas lausevormis, mis sisaldab alamsõna α, asendada selle sõnaga β. DF: Generatiivseks grammatikaks e. lihtsalt grammatikaks nim. nelikut $G = (\Sigma,N,P,S_0)$, kus: 1.Σ on terminaalide tähestik, 2.N on mitteterminaalide tähestik, kusjuures $\Sigma\cap N = \emptyset$, 3.$P\subseteq V^*NV^*\times V^*$ on prod.ide hulk, kus $V = \Sigma\cup N$, 4.$S_0\subseteq N$ on grammatika lähtesümbol. DF: Lausevorm ψ on grammatikas G vahetult tuletatav lausevormist φ, kui $\varphi = \gamma\alpha\delta$ ja $\psi = \gamma\beta\delta$ $(\gamma,\beta\in V^*)$ ning grammatika G prod.ide hulk P sisaldab prod.i $\alpha\to\beta$. Vahetu tuletatavus on binaarne relatsioon stringide hulgal V*, mida tähist. $\Rightarrow_G$. DF: Grammatika G poolt genereeitav keel on hulk $L(G) = \{w|w\in\Sigma^*, s\Rightarrow^* w\}$. Väide: $L(G) = \{ww|w\in\{a,b\}^*\}$. DF: Grammatikat G = (S,N,P,S) nim.: 1.0-tüüpi grammatikaks L_0, kui talle pole seatud mingeid lisakitsendusi, 2.1.-tüüpi grammatikaks L_1 e. kontekstist sõltuvaks grammatikaks, kui ∀ prod.i $\alpha\to\beta\in P$ korral kehtib seos $0<|\alpha|<=|\beta|$, v.a. juhul kui $\epsilon\in L(G)$, siis sisaldab prod.ide hulk ühe elemendi, mis ei rahulda eelpool toodud seost: $S\to\epsilon$. 3.2.-tüüpi grammatikakas L_2 e. kontekstivabaks (KV) grammatikaks, kui ∀ prod. on kujul $A\to w$, kus $A\in N$ ja $w\in V^*$, 4.3.-tüüpi grammatikaks L_3 e. paremlineaarseks grammatikaks, kui kõik prod.id on kujul $A\to bC$ või $A\to b$, kus $A,C\in N$, $b\in\Sigma$ või $b = \epsilon$. Järeldus viimasest definitsioonist: Omadus: $L_0\supseteq L_1\supseteq L_2\supseteq L_3$.
Chomsky klassifikatsioon.
8.Regulaarsed avaldised ja hulgad. DF: Olgu Σ lõplik tähestik. Regulaarseks hulgaks tähestikuks Σ nim.: 1.tühja hulka ∅, 2.hulka, mis koosneb vaid tühjast sõnast $\{\epsilon\}$, 3.{a}, kus $a\in\Sigma$, 4.hulki $P\cup Q$, PQ ja Q^*, kui P ja Q on regulaarsed hulgad. Regulaarsete hulkade mugavamaks tähistamiseks kasut. nn. regulaarseid avaldisi. DF: Regulaarne avaldis on määratletud järgmiste rekursiivsete reeglitega: 1.∅ on regulaarne avaldis, mis tähistab tühja hulka, 2.e on reg.avaldis, mis täh. hulka $\{\epsilon\}$, 3.a on reg.avaldis, mis täh. hulka {a}, kus $a\in\Sigma$, 4.kui p ja q on reg.avaldised, mis tähistavad vastavalt regulaarseid hulki P ja Q, siis on reg.avaldised ka: a.(p+q), mis täh. hulka $P\cup Q$, b.(pq), mis tähistab hulka PQ, c.$(p)^*$, mis tähistab hulka P^*, 5.muud avaldised ei ole regulaarsed. Regulaarse avaldise tehetest on kõrgeima prioriteediga operatsioon , seejärel tuleb konkatenatsioon ja siis operatsioon +. Avaldis p^+ on avaldise pp^* lühend. Regulaarseid avaldisi loetakse võrdseiks, kui nad tähistavad üht ja sama hulka. LM: Regulaarsed hulgad ∅, $\{\epsilon\}$ ja {a}, kus $a\in\Sigma$, on paremlineaarsed keeled. LM: Kui L_1 ja L_2 on paremlineaarsed keeled tähestikus Σ, siis on paremlineaarsed keeled ka: 1.$L_1\cup L_2$, 2.L_1L_2, 3.L_1^*. Nende lemmadega järelduseks on Teoreem: Regulaarne hulk on genereeritav paremlineaarse grammatikaga.
9.Lõplikud automaadid. Grammatika või regulaarse avaldisega esitatud keele lausete genereerimise ülesandega duaalne ülesanne on lausete aktsepteerimise ülesanne. Regulaarsete hulkade aktsepteerimise ülesande lahendamiseks võib kasutada lõplikku automaati. DF: Lõplik automaat on viisik $M = (\Sigma,Q,\delta,Q_0,F)$, kus: 1.Σ on sisendtähestik, 2.Q

on olekusümbolite lõplik tähestik, 3.$\delta:\Sigma\times Q\rightarrow P(Q)$ on üleminekufunktsioon, 4.$Q_0\subseteq Q$ on lähteolekute hulk, 5.$F\subseteq Q$ on lõppolekute hulk. **Mittedeterministlike automaatide teisendamine deterministlikeks.** Kui $|\delta(a,q)| = 1$ $\forall$ $a\in\Sigma$ ja $q\in Q$ korral ning $Q_0 = \{q_0\}$, nim. lõplikku automaati M deterministlikuks, vastasel juhul mittedeterministlikuks. Lõpliku automaadi $M = (\Sigma,Q,\delta,Q_0,F)$ konfiguratsiooniks nim. paari $(w,q)\in\Sigma^*\times Q$. Konfiguratsiooni kujul (w,q_0), kus $q_0\in Q_0$ nim.lõpliku automaadi M lähtekonfiguratsiooniks ning konfiguratsiooni (ϵ,r), kus $r\in F$ lõppkonfiguratsiooniks. DF: Lõpliku automaadi töötaktiks nim. binaarset seost |- konfiguratsioonide hulgal: (aw,q_1) |- (w,q_2) kehtib siis, kui $q_2\in\delta(a,q_1)$. DF: Lõpliku automaadi $M = (\Sigma,Q,\delta,Q_0,F)$ poolt aktsepteeritav keel $T(M) = \{w|(w,q_0) \vdash^* (\epsilon,r), q_0\in Q_0, r\in F\}$. TR: $\forall$ lõpliku automaadi poolt aktsepteeritav keel on paremlineaarne. TR: $\forall$ lõpliku automaadi poolt aktsepteeritav keel on regulaarne hulk. TR: $\forall$ paremlineaarne keel on aktsepteeritav (mittedeterministliku) lõpliku automaadi abil. TR: $\forall$ mittedeterministliku lõpliku automaadi N jaoks $\exists$ selline deterministlik lõplik automaat M, nii et $T(M) = T(N)$.

10.Regulaarsete avaldiste, lineaarsete grammatikate ja lõplike automaatide samaväärsus.
TR: Ekvivalentsed on väited: 1.L on regulaarne hulk, 2.L on paremlineaarne keel, 3.L on aktsepteeritav deterministliku lõpliku automaadi abil.

11.Keele regulaarsuse tarvilik tingimus. TR: Tarvilikkuse tingimus keele regulaarsuseks: Olgu deterministlik lõplik automaat, millel on n olekut. $\forall$ sõna $z\in T(M)$, mille korral $|z|\geq n$ on esitatav kujul $z = unw$, nii et $\forall$ $j\geq 0$ korral $uv^jw\in T(M)$.

12.KV-keelte süntaksi- ja tuletuspuud. KV-keelte omaduste analüüsilmisel kasut. tihti süntaksipuid. $\forall$ järjestatud puu $T = (A,R)$, mille tippude märgistus on antud kujutusega f, on esitatav termina: 1.kui $a\in A$ on puu T terminaalne tipp, siis märgend $M = f(a)$ on term, 2.kui $a\in A$ on puu T mitteterminaalne tipp märgendiga $M = f(a)$, mille vahetuid alampuid tähistavad termid $t_1,t_2,\ldots,t_n$, siis $M(t_1,t_2,\ldots,t_n)$ on term, mis tähistab puu alampuud juurega a. Toodud term on saadav puu eesjärjekorrast lrep(T), asendade selles tipud neile vastavate märgenditega. Puu krooniks Kr(T) nim. stringi, mis saadakse tema terminaalsete tippude märgenditest, mis on kirjutatud järjestikku alates kõige vasakpoolsemast. Puu termist saab tema krooni, kui termist kustutada kõik need märgendid, mille järgneb avanev sulg ning seejärel kõik sulud ja komad. Kui meid ei huvita puu sisemised tipud, kasutame tema tähistusena termi A[Kr(t)], kus A on puu juure t märgend. Kui asendame puus t_1 terminaalse tipu A puuga T_2, mille juur on märgendatud sümboliga A, tähistame $t = t_1\{A/t_2\}$. Kui $t_1 = S[\alpha A\gamma]$ ja $t_2 = A[\beta]$, siis $t = S[\alpha\beta\gamma]$. Puud, mis koosneb vaid juurest ja terminaalsetest tippudest, nim. elementaarpuuks. Iga puu on üheselt esitatav oma elementaarpuude nimistuga E(T) (korteez). Korteezidevaheliste ning korteezide ja elementide vaheliste kuuluvusseoste esitamiseks kasut. hulgateooria tähiseid. Märgendatuid järjestatud elementaarpuid võib kasutada KV-G prod.ide esitamiseks: puu $A(x_1,x_2,\ldots x_n)$ kujutagu prod.i $A\rightarrow x_1,x_2,\ldots x_n$. Kui elementaarpuu r esitab prod.i p, siis tähistatakse seda $r|\rightarrow p$. DF: Süntaksipuuks KV-Gs $G = (\Sigma,N,P,S)$ nim. märgendatud järjestatud puud t, kui $\forall$ $r\in E(t)$ korral $R|\rightarrow p$, $p\in P$. TR: KV-Gs $G = (\Sigma,N,P,S)$ on mitteterminaalist $A\in N$ tuletatav lausevorm α siis, kui grammatikas G $\exists$ süntaksipuu $A[\alpha]$. DF: Täielikku süntaksipuud t, mille kroon $Kr(t) = x\in\Sigma^*$ nim. sõna x tuletuspuuks. Täielik on süntaksipuu, mida ei saa laiendada, tema juur on märgendatud grammatika lähtesümboliga ja kõik lehed on märgendatud terminaalidega. Tuletuspuud. Sõna x kuulub keelde L(G) siis, kui grammatikas G $\exists$ tuletuspuu t nii, et Kr(t) = x. Tuletuspuu näitab, millistes grammatilistes seostes on genereeritava lause liikmed ning ta on ka lause semantika väljendajaks. Kui lausele saab ehitada mitu tuletuspuud, on lausel ka mitu erinevat tähendust. DF: KV-Gt G, mille korral $\exists$ sõnal $x\in L(G)$ mitu erinevat tuletuspuud, nim. mitmeseks. DF: KV-keelt L, millel $\exists$ ühene genereeriv grammatika G ($L = L(G)$), nim. üheseks, vastasel juhul mitmeseks. DF: KV-G G on ühene, kui ei leidu sõna $x\in L(G)$, nii et tema erinevad vasak- või paremtuletused omaksid erinevaid tuletuspuid.

13.KV-grammatikate redutseerimine. On olemas viis algoritmi: A1:kas keel L(G) on tühi?, A2:saavutamatute sümbolite elimineerimine, A3:kasutute sümbolite elimineerimine, A4:ϵ-reeglite elimineerimine, A5:ahelprod.ide elimineerimine. Need algoritmid võimaldavad KV-Gt lihtsustada ja viia standarsele kujul. ϵ-reegliks nim. prod.i $A\rightarrow\epsilon$. KV-Gt $G = (\Sigma,N,P,S)$, nim. ϵ-vabaks, kui: 1.P ei sisalda ϵ-reegleid, 2.P sisaldab ühe S-reegli $S\rightarrow\epsilon$ ning lähtesümbol S ei sisaldu ühegi prod.i paremas pooles.
DF: KV-Gkt $G = (\Sigma,N,P,S)$ nim. redutseerituks, kui ta ei sisalda tsükleid, ϵ-reegleid ja kasutuid sümboleid. TR: $\forall$ KV-keele L jaoks $\exists$ selline redutseeritud KV-G G, et $L(G) = L$.

14.KV-grammatikate normaalkujud. Olgu KV-Gs $G = (\Sigma,N,P,S)$ prod. $A\rightarrow\alpha B\beta$, kus $B\in N$ ja $\alpha,\beta\in(\Sigma\cup N)^*$. Eeldame, et hulk P sisaldab $B\rightarrow\gamma_1,\ldots,B\rightarrow\gamma_k$. Moodustame uue KV-G $G' = (\Sigma,N,P',S)$, kus $P' = (P\backslash\{A\rightarrow\alpha B\beta\})\cup\{A\rightarrow\alpha\gamma_1\beta,\ldots,A\rightarrow\alpha\gamma_k\beta\}$. LM: $L(G) = L(G')$. DF: KV-G $G = (\Sigma,N,P,S)$ on Chomsky normaalkujul, kui $\forall$ prod. hulgas P on ühel järgnevatest kujudest: 1.$A\rightarrow BC$, kus $A,B,C\in N$, 2. $A\rightarrow a$, kus $a\in\Sigma$, 3.$S\rightarrow\epsilon$, kui $\epsilon\in L(G)$ ja grammatika lähtesümbol S ei esine prod.ide paremal poolel. AL: Grammatika teisendamine Chomsky normaalkujule. TR: $\forall$ KV-G G on teisendatav temaga ekvivalentseks grammatikaks G' Chomsky normaalkujul, kasutades vastavat algoritmi. DF: ϵ-vaba KV-G $G = (\Sigma,N,P,S)$ on Greibachi normaalkujul, kui kõik prod.id on kujul $A\rightarrow a\alpha$, kus $a\in\Sigma$ ja $\alpha\in N^*$. Mitteterminaal A nim. rekursiivseks, kui $A\Rightarrow^+\alpha A\beta$. Kui $\alpha=\epsilon$, siis on A on vasakrekursiivne, ku $\beta=\epsilon$, siis A on paremrekursiivne. Prod.i kujul $A\rightarrow\alpha$ nim. A-prod.iks. LM: Olgu $G = (\Sigma,N,P,S)$ KV-G, mille kõikideks A-prod.ideks on $A\rightarrow A\alpha_1,\ldots, A\rightarrow A\alpha_m, A\rightarrow\beta_1,\ldots, A\rightarrow\beta_n$. Olgu $G' = (\Sigma,N\cup\{A'\},P',S)$, kus A' on uus mitteterminaal ning P' on saadud hulgast P kõigi A-prod.ide asendamisel prod.idega $A\rightarrow\beta_1,\ldots, A\rightarrow\beta_n$,

$A\to\beta_1A'$,…, $A\to\beta_nA'$, $A'\to\alpha_1$,…, $A'\to\alpha_m$, $A'\to\alpha_1A'$,…, $A'\to\alpha_mA'$. Siis $L(G') = L(G)$. AL: vasakrekursiooni elimineerimine. TR: $\forall$ KV-keel on genereeritav mitte-vasakrekursiivse grammatikaga. LM: Kui KV-G G = (Σ,N,P,S) on mittevasakrekursiiivne, siis $\exists$ mitteterminaalide hulgal N järjestus, nii et $A\to B\alpha\in P \Rightarrow A<B$. AL: KV-G viimine Greibachi normaalkujule. TR: $\forall$ keele L jaoks $\exists$ grammatika G Greibachi normaalkujul, nii et $L(G) = L$.

15.KV-keelte süntaksanalüüs. Süntaksanalüüsi ülesande korral on antud keelt genereeriva grammatika G põhjal vaja otsustada, kas sõna w kuulub keelde L(G) või mitte. Earley' algoritm püüab süntaksanalüüsi ülesande lahendamiseks ehitada sõna w vasaktuletust. Grammatika G prod.i $A\to v$ võib olla sõna $w = x_1...x_n$, kus $x_j\in\Sigma$, vasaktuletuses kasutatud siis, kui $\exists$ tuletus: $S \Rightarrow^* x_1\ldots x_nA\delta$. Oletame, et sõna v koosneb alamsõnadest α ja β ($v=\alpha\beta$). Sellise täpsustuse korral saab prod.i $A\to v$ kasutatavuse tervikliku tingimuse esitada täpsemalt: $S \Rightarrow^* x_1\ldots x_iA\delta \Rightarrow x_1\ldots x_i\alpha\beta\delta \Rightarrow^* x_1\ldots x_ix_{i+1}\ldots x_j\beta\delta$. Earley' algoritmis moodustatakse massiiv $||I_{ij}||$, mille elementideks on nn. punktiga prod.ide hulgad. Prod. $A\to\alpha\cdot\beta$ kuulub elemendi I_{ij} koosseisu siis, kui sõna $w = x_1\ldots x_n$ vasaktuletuse saab ehitada osatuletuse jätkamise teel. Sõna w kuulub keelde $L(G)$, kui prod. $S\to v.$ kuulub elementi $I_{0,n}$. LM: Olgu $B\to.v\in I_{ii}$ ning eeldame, et $v \Rightarrow^* x_{i+1},\ldots,x_j$. Siis $B\to.v \in I_{ij}$. LM: Olgu $A\to w_1\cdot w_2w_3\in I_{ik}$, kus $w_1,w_2,w_3\in(\Sigma\cup N)^*$ ja oletame, et $w_2 \Rightarrow^* x_{k+1}\ldots x_j$,siis $A\to w_1w_2\cdot w_3\in I_{ij}$. LM: Olgu $B \Rightarrow w \Rightarrow^* w_1Aw_2$, kus $w,w_1,w_2\in(N\cup\Sigma)^*$ ning $B\to.w\in I_{kk}$ ja $w \Rightarrow x_{k+2}\ldots x_j$. Siis $\forall$ A-prod.i $A\to Z$ korral $A\to.Z\in I_{jj}$. TR: (Earley' algoritmi korrektsus) Etteantud stringi $w = x_1\ldots x_n\in\Sigma^*$ korral paigutub algoritm prod.i $A\to\alpha.\beta$ elementi I_{ij} siis, kui grammatikas $\exists$ vasaktuletus: $S \Rightarrow^* x_1\ldots x_iA\delta \Rightarrow x_1\ldots x_i\alpha\beta\delta \Rightarrow^* x_1\ldots x_j\beta\delta$. Erijuhul kui $S\to v.\in I_{0,n}$, siis $w\in L(G)$. TR: Earley' algoritmi ajaline keerukus on $O(n^3)$, kus n on analüüsitava sõna pikkus. Juhul kui KV-G G on ühene, on ajaline keerukus $O(n^2)$.

CKY-algoritm. Cocke - Kasami - Youngeri algoritm. CKY algoritm lahendab süntaksanalüüsi ülesande KV-Gte korral, mis on Chomsky normaalkujul. Sellisel juhul analüüsitava sõna süntaksipuu on kahendpuu. Tuletuspuude konstrueerimisel võib tekkida mitu erinevat tuletuspuud. CKY algoritm ühendab erinevad võimalikud tuletuspuud, mis on ühe või teise analüüsitava stringi alamsõna süntaksipuudeks. CKY algoritm täidab püramiidikujulise tabeli, mille alus koosneb n elemendist, selle kohal on n-1 elementi jne. Tabeli lahtrisse aadressiga (i,j) paigutatakse mitteterminaal A siis, kui $A \Rightarrow^* x_jx_{j+1}\ldots x_{j+1-1}$ ehk A on aadressil (i,j), kui talle alluva püramiidi alus toetub sõnale $x_jx_{j+1}\ldots x_{j+1-1}$. TR: Algoritmi tulemusena $A\in E(i,j)$ siis, kui grammatikas G on esitatav tuletus $A\to x_jx_{j+1}$. $w\in L(G)$ siis, kui $S\in E(n,1)$. TR:CKY-algoritmi ajaline keerukus on $O(n^3)$, kus n on analüüsitava sõna pikkus.

16.Magasinmäluga automaadid. MMA on seadeldis KV-keelte aktsepteerimiseks. Automaadil on sisendlint ja mälu. Sisendlindil olevaid sümboleid saab automaat ühekaupa lugeda. Mälu on organiseeritud magasinina, kuhu salvestatakse automaadi töö käigus läbitud olekud. Töö algul on automaadi magasin tühi. Masina tööd juhib programm, mis näitab, kuidas muuta mälu seisu lindilt järjekordse sümboli lugemisel koos samaaegse sümboli lugemisega magasinist. DF: MMA on struktuur $M = (\Sigma,\Gamma,Q,p,q_0,\$,F)$, kus Σ on sisendtähestik, Γ on magasini tähestik, Q on olekute tähestik, $p:(\Sigma\cup\{\epsilon\}\times\Gamma\times Q\to\Gamma^*\times 2^Q$ on üleminekufun., $q_0\in Q$ on lähteolek, $\$\in\Gamma$ on magasini lähtesümbol, $F\subseteq Q$ on lõppolekute hulk. MMA $M = (\Sigma,\Gamma,Q,p,q_0,\$,F)$ konfiguratsiooniks nim. kolmikut $(w,\gamma,q)\in\Sigma^*\times\Gamma\times Q$, kus w esitab sisendlindil veel lugemata alamstringi, γ magasini mälus salvestatud stringi ja q automaadi jooksvat olekut. MMA M lähtekonfiguratsiooniks nim. kolmikut $(w,\$,q_0)$ ja lõppkonfiguratsiooniks (ϵ,ϵ,r), kus $r\in F$. DF: MMA töötaktiks nim. binaarset seost |- konfiguratsioonide hulgal, nii et $(aw,Z\alpha,q)$ |- $(w,\gamma\alpha,q')$, kui $(\gamma,q')\in p(a,Z,q)$ või $(w,Z\alpha,q)$ |- $(w,\gamma\alpha,q')$, kui $(\gamma,q')\in p(\epsilon,Z,q)$. DF: MMA poolt aktsepteeritav keel on hulk $T(M) = \{w|(w,\$,q_0) \vdash^* (\epsilon,\epsilon,r)$, kus $r\in F\}$. TR: $\forall$ KV-keele L jaoks $\exists$ MMA M nii, et L = T(M). DF: MMA M' ja M'' on ekvivalentsed, kui T(M') = T(M'').

17.Ühe olekuga magasinmäluga automaatide ja Greibachi mõttes normaliseeritud KV-grammatikate ekvivalentsus. LM: $\forall$ MMA M jaoks $\exists$ ühe olekuga MMA M' nii, et T(M) = T(M'). TR: $\forall$ MMA M jaoks $\exists$ KV-G G nii, et $T(M) = L(G)$.

18.KV-keelte tarvilikkuse tingimus. Süntaksipuu kõrguseks nim. temas leiduva pikima tee kaarte arvu. LM: Tähistame KV-Gs G = (Σ,N,P,S) sümboliga $m = \max_{A\to w\in P}|w|$. Olgu sõna x tuletuspuu kõrgus j. Siis $|x|<=m^j$. TR: KV-keelete tarvilikkuse tingimus: Olgu KV-keel L. Siis leiduvad ainult keelest L sõltuvad konstandid p ja q, nii et $z\in L$ & $|z|>p$ korral $\exists$ jaotus z = uvwxy, nii et: 1.|vwx|<=q, 2.v ja x pole korraga tühjad sõnad, 3.$\forall$ $i\geq 0$ korral $uv^iwx^iy\in L$. JR: Sisalduvus $L_2\subset L_1$ on range.

19.Atribuutgrammatika, programmi semantika leidmine. Programmeerimiskeelte semantika realiseerimiseks kasut. atribuuttehnikat. Üldjuhul esitab lause mingi seose teatud objektide vahel. Programmeerimiskeele korral on neiks objektideks arvutis salvestatavad ning töödeldavad andmeobjektid. Objektide kogumit nim. keele semantiliseks piirkonnaks. Keele L lause semantika spetsifitseerimiseks tuleb kirjeldada vastavus keele lausete (KV-keelete puhul lausete tuletuspuude) ja semantilise piirkonna D objektide vahel: $\varphi:L\to D$. Semantilise piirkonna D kirjeldamiseks kasut. semantilisi muutujaid hulgast $X = \{x_1,x_2,\ldots\}$, millest igaühe jaoks on def. muutumispiirkond e. doomen hulgas D. Semantlise muutuja x_i jaoks määratud võimalike väärtuste hulk $D(x_i) = Dx_i\subseteq D$. Iga semantilise muutuja väärtuste hulk peab olema osaliselt järjestatud ning sisaldama täielkult määramata väärtuse "nil". Kui muutuja x jooksvaks väärtuseks on $nil\in D_x$, tähendab see, et seni pole muutuja väärtust veel arvutatud. Kujutuse φ määratlemiseks kasut. atribuuttehnikas formalismi, mida nim. atribuutgrammatikaks. DF: Atr.G on struktuur AG = (G,A,R), kus $G = (\Sigma,N,P,S_0)$ on KV-G, $A\to P(X)$ on KV-G G sümbolite hulga $A = \Sigma\cup N$ atribuutide hulk, $R = \{R_p\}_{p\in P}$ on semantikareeglite pere. Iga KV-G G prod.i semantikareegel R_p on omistamiste hulk, mis määrab, kuidas arvutada tuletuspuu atribuute. DF: Prod. $p:X_0 \rightarrow X_1 \ldots X_{np}$ semantikareegel on omistamiste hulk $R_p = \{X_0.s := f_s(X_0.a_1,\ldots,X_{np}.d_n) | s \in S(X_0)\} \cup \{X_k.i := f_i(X_0.a_1,\ldots,X_{np}.d_n) | 0<k<=n_p,\ i \in I(X_k)\}$, kus f_s ja f_i on antud semantilises piirkonnas määratud avaldised. Atr.G määrab reeglid, kuidas arvutada sõna $x \in L(G)$ semantikat. Tuleb arvutada sõna x tuletuspuu tippude atribuudid. Vastavat asrvutusprotsessi nim. tuletuspuu dekodeerimiseks. Dekodeerimine on protsess, mis seni, kuni saab arvutada uusi atribuutide väärtusi, täidab tsükliliselt teatud tingimuslikku operaatorit. Dekodeerimise lähteseisus olgu puu kõigi atribuutide väärtus "nil". DF: Atr.G on korrektselt määratud, kui $\forall$ sõna $x \in L(G)$ korral on dekodeeritud tuletuspuus kõigi atribuutide väärtused määratud e. $x \neq$nil. DF: Sõna $x \in L(G)$ semantika on väärtus $\varphi(X,K_1,\ldots,K_m) = <Z_1,\ldots,Z_n>$, kus $Z_1,\ldots,Z_n$ on sõna x dekodeeritud tuletuspuu juure sünteesitud atribuutide väärtused ning $K_1,\ldots,K_m$ puu juure päritud atribuutide väärtused. Atr.G esitab semantika transleeritava sõna struktuuri ja konteksti põhjal.

20.Turingi masin. TM on abstraknte mudel, mis esitab matem. täpselt algoritmi ja arvutusprotsessi mõisted ning võimaldab neid matem. meetoditega uurida. Turingi masinat võib ette kujutada koosnevana arvutusseadmest ja lindist info kodeeritud esitamiseks. Lint on mõlemas suunas lõpmatu ning jaotatud pesadeks, millest igaüks võib sisaldada ühe sümboli kindlast tähestikust. Lugemis-kirjutamispea võib teatud kindla aja jooksul lugeda lindilt ühe sümboli või salvestada sinna ühe sümboli. Kirjutamisel kirjutatakse varem samas pesas olnud sümbol üle. Arvutusseade saab ühe ajakvandi jooksul liigutada l/k-pead ühe pesa võrra paremale või vasakule. Eeldatakse ka veel, et passivses olekus on TMa l/k-pea selle pesa kohal, kus algab kasulik info. DF: TM on viisik $A = (A_t,Q,p,q_0,Q_f)$, kus: 1.A_t on sümbolite lõplik tähestik, Q on olekute lõplik tähestik, p on üleminekufun.,q_0 on lähteolek, Q_f on lõppolekute hulk. DF: $\gamma \in A_t^* \times Q \times A_t^*$ on TMa A konfiguratsioon. Lähtekonf. on q_0x ja lõppkonf. on ry . **Lahenduvad ja genereeritavad hulgad.** DF: TMat $A(A_t,Q,p,q_0,Q_f)$, kus $Q_f = \{q_t,q_f\}$ nim. hulga $X \subseteq At^*$ karakterislikuks TMaks, kui $A(x) \rightarrow q_t$, kui $x \in X$, $A \rightarrow q_f$, kui $x \notin X$. Hulka, mille jaoks leidub kar.TM, nim. lahenduvaks hulgaks. Hulka $M(A=\{x | A(x)<\infty\})$ nim. TMa määramispiirk.-ks. Hulka, mis on mingi TMa määr.piirk.-ks, nim. genereeritavaks hulgaks.

21.Turingi masina kodeerimine. TMa käsk on sõna $q_i a_j q_k X$, kus q_i on jooksev olek, a_j on loetav sümbol, q_k on järgmine olek, X on tegevus ($X \in A_t \cup \{L,R\}$). L tähendab l/k-pea nihutamist vasakule ja R paremale. TMa üleminekufun. võib olla esitatud programmina e. käskude jadana. Käsud on jadas üksteisest eraldatud sümbolitega ";". Iga TMat saab kodeerida, esitades ta sõnana tähestikus $A_2 = \{\_,0,|\}$. **Algoritmiliselt mittelahenduvad ülesanded.** DF: TMat U, mille korral suvalise TMa R ja argumendi x puhul kehtib seos $U(R(U)*(R(x)) = R(U(x))$, nim. universaalseks TMaks. TR: Ei leidu sellist TMat, mis suvalise TMa A ja lähtesõna x korral tunneb ära, kas masin lõpetab töö või mitte. JR: Leidub ülesandeid, mis pole algoritmiliselt lahenduvad. JR: Univ. TMa määramispiirk. ei ole lahenduv hulk. **Church-Turingi tees.** TM tähtsus põhineb vastaval teesil: iga efektiivselt arvutava funktsiooni võib realiseerida Turingi masinal

22.Lihtrekursiivsed funktsioonid, nende arvutatavus Turingi mõttes. Operaatorid: 1.Superpositsioon – fun.de järjest rakendamine. 2.Lähendamine rekursiooni abil, kusjuures kasut. sama funktsiooni eelmisi väärtusi. 3.Minimeerimine – arvutatakse niikaua, kuni on täidetud teatud tingimus. DF: n-kohaline fun. f on saadud m-kohalisest fun.-st g ning n-kohalistest fun.-dest $h_1,\ldots,h_m$ superpositsiooni operaatori S^{m+1} rakendamise teel [$f = S^{m+1}(g; h_1,\ldots,h_m)$], kui kõikide $x_1,\ldots,x_n$ väärtuste korral kehtib: $f(x_1,\ldots,x_n) = g(h_1(x_1,\ldots,x_n),\ldots, h_m(x_1,\ldots,x_n))$. DF: (n+1)-kohaline fun. f on saadud n-kohalisest fun.-st g ning (n+2)-kohalisest fun.-st h rekursiooni operaatori R rakendamise teel [$f=R(g,h)$], kui kõikide $x_1,\ldots,x_n,y$ väärtuste korral kehtivad: $\{1.f(x_1,\ldots,x_n,0) = g(x_1,\ldots,x_n)$, 2. $f(x_1,\ldots,x_n,y+1) = h(x_1,\ldots,x_n,y, f(x_1,\ldots,x_n,y))$. DF: Funktsioone $O^n(x_1,\ldots,x_n) = 0$, $s(x) = x+1$, $I_m^n(x_1,\ldots,x_n) = x_m$ nim. algfunktsioonideks. TR: Algfunktsioonid on arvutatavad Turingi mõttes. DF: Fun. $f: N^n \rightarrow N$ nim. lihtrekursiivseks (LRF), kui f on algfun. või ta on saadav algfunktsioonidest superpositsiooni ja rekursioonioperaatori abil.

23.Minimeerimisoperaator. DF: n-kohaline fun. f on saadud n+1-kohalisest fun.-st g minimeerimisoperaator μ_y abil ning kirjutatakse $f = \mu_y[g]$, kui muutujate $x_1,\ldots,x_n$ kõigi väärtuste korral $f(x_1,\ldots,x_n) = y$, kus y on vähim selline element, et $g(x_1,\ldots,x_n,y) = 0$, kusjuures iga z<y korral $g(x_1,\ldots,x_n,z)$ on määratud ja $\neq 0$; vastasel juhul ei ole määratud. **Osaliselt rekursiivsed funktsioonid.** Funktsioone, mis on saadud algfun.-dest superpostsiooni-, lihtrekursiooni- ja minimeerimisoperaatori rakendamise teel, nim. osaliselt rekursiivseteks fun.-deks. Lihtrekursiivsed fun.-id on osaliselt rekursiivsed. Osaliselt rekursiivsed fun.-id ei pruugi olla kõikjal määratud. DF: Kõikjal määratud osaliselt rekursiivseid fun. nim. üldrekursiivseteks fun.-deks. Osaliselt rekursiivsete fun.-de hulk langeb kokku Turingi mõttes arvutatavate fun.-de hulgaga.

24.Cantori funktsioonid. TR: Cantori TR: Kui asenduste suhtes kinnises kõikjal määratud fun.-de klassis $\exists$ selline fun. g, et iga väärtuse z korral $g(z) \neq z$, siis ei sisalda F univ.fun. alamklassile F^k ühegi k=1,2,… korral. Kui fun.-de klass F sisaldab Cantori fun. ja selle pöördfun.-e, siis saab ka alamklassi F^k universaalse fun. asendada ühekohaliste fun.-de klassi F^1 univ. fun.-ga: $U^k(a,x_1,\ldots,x_n) = U^1(a,c^k(x_1,\ldots,x_n))$. **Arvutatava funktsiooni ühekohalised esindajad.** TR: Kui superpositsioonioperaatori suhtes kinnine fun.de klass sisaldab korteezide kodeerimise ja dekodeerimise fun.-e $c_m,c_1^m,\ldots,c_m^m$, siis selle klassi $\forall$ m-kohalise fun. f jaoks $\exists$ samasse klassi kuuluv ühekohaline fun. (fun. ühekohaline esindaja) g nii, et muutujate $x_1,\ldots,x_m$ kõigi fun. f määramispiirk. kuuluvate väärtuste korral kehtib: $f(x_1,\ldots,x_m) = g(c^m(x_1,\ldots,x_m))$.

25.Arvutatavate funktsioonide klassi universaalne funktsioon. DF: k+1-kohalist fun. U nim. alamklassi F^k universaalseks fun.-ks, kui: 1.$\forall$ fiks. väärtuse a korral kuulub muutujatest $x_1,\ldots,x_n$ sõltuv fun. $U(a,x_1,\ldots,x_k)$ klassi

F^k; 2.$\forall$ fun. $f\in F^k$ korral $\exists$ selline nat.arv b, et kõikide muutujate $x_1,\dots,x_n$ korral kehtib $f(x_1,\dots,x_k) = U(b,x_1,\dots,x_n)$. Kui fun.-de klass F sisaldab Cantori fun. ja selle pöördfun.-e, siis saab ka alamklassi F^k universaalse fun. asendada ühekohaliste fun.-de klassi F^1 univ. fun.-ga: $U^k(a,x_1,\dots,x_n) = U^1(a,c^k(x_1,\dots,x_n))$.

26.Ühekohaliste funktsioonide arvutatavus. Mitmekohalisi rekursiivseid fun. võib käsitleda ka ühekohalistena. Selleks tuuakse sisse korteezide järjestus. Kahekohaliste korteezide järjekorra moodustamiseks sobib Cantori fun. $c(x,y) = 0{,}5(x+y)(x+y+1)+x$. TR: Robinsoni I TR: Kõik ühekohalised lihtrekursiivsed fun.id on saadavad algfun.-dest $s(x) = x+1$& $q(x) = x-\lfloor\sqrt{x}\rfloor^2$ kasutades liitmise, kompositsiooni ja iteratsioonioperaatorit. TR: Robinsoni II TR: Kõik ühekohalised osaliselt rekursiivsed fun.id on saadavad algfun.-dest $s(x) = x+1$ & $q(x) = x-\lfloor\sqrt{x}\rfloor^2$ kasutades liitmise, kompositsiooni ja pööramisoperaatorit. **Gödeli numbrid.** Fun. gödeliseerimiseks nim. tema Gödeli numbri väljaarvutamist. DF: Fun. h Gödeli number on arv Gh, mis arvutatakse nii: Gh = 2, kui h=s; 3, kui h=q; $5^{Gf}*7^{Gg}$,kui h=f+g; $11^{Gf}*13^{Gg}$,kui h=f•g; 17^{Gf},kui $h=f^{-1}$; 19^{Gf},kui h=ιf. Sisuliselt kodeerib Gödeli number fun.-le vastava operaatortermi.

27.Kleene' s-n-m-teoreem ja püsipunktiprintsiip. TR: Kleene' s-m-n TR: $\exists$ selline m+1-kohaline arvutatav fun. S_n^m nii, et suvaliste $a,y_1,\dots,y_m$ korral kehtib: $\varphi^{(n)}[S_n^m(a,y_1,\dots,y_m) = \lambda z_1,\dots,\lambda z_n.\varphi_a^{(m+n)}(y_1,\dots,y_m, z_1,\dots,z_n)$. S-m-n TR järeldus on järgmine TR. TR: Kleene' püsipunktiprintsiip: mistahes arvutatava fun. h korral leidub selline nat.arv u, et kehtib: $\varphi_u = \varphi_{h(u)}$. Antud TR nim. ka rekursiooni TR-ks. Fun. $\varphi_{h(u)}$ kujutab endast arvutatavate fun.-de superpositsiooni.

28.Rice'i teoreem. TR: Arvutatavate fun.-de Gödeli numbrite iga mittetriviaalne hulk on mittelahenduv. Mittetriviaalseks loetakse siin mittetühja ja mitteuniversaalset arvutatavate fun.-de hulka.

29.Formaalsed loogika arvutused, tuletuse mõiste.

30.Sekventsiaalloogika.

31.Valemi disjunktide hulga moodustamine.

32.Resolutsiooniprintsiip ja keel PROLOG:

33.Intuitsionistlik lausearvutus.

34.Programmide sünteesi loogilised alused.