**Лекция № 3.** Логикалық келтірілетін Тьюринг машиналары. Тьюринг әмбебап машинасы. Программа дұрыстығын дәлелдеу әдістемесі.

**МАҚСАТЫ:** логикалық келтірілетін тьюринг машиналары мен тьюринг әмбебап машинасы, программа дұрыстығын дәлелдеу әдістемесі мен қиындық функциясы негізінде алгоритмдерді классификациялау. есептеліну және шешімі болу ұғымдарымен таныстыру.

**СҰРАҚТАР**:

1. Логикалық келтірілетін Тьюринг машиналары.
2. Тьюринг әмбебап машинасы.
3. Программа дұрыстығын дәлелдеу әдістемесі.
4. Қиындық функциясы негізінде алгоритмдерді классификациялау.
5. Есептеліну және шешімі болу.

**НЕГІЗГІ ҰҒЫМДАР:** Логикалық келтірілетін Тьюринг машиналары. Тьюринг әмбебап машинасы. Программа дұрыстығын дәлелдеу әдістемесі.

**Логикалық қайтымды және қайтымсыз компьютерлер**

Қазіргі таңда есептеу теорияларын физикадан бөліп қарауға болмайды. (бүгінде «Тьюрингтің кванттың машинасына» жүгінеді, ал «q-бит» термині «бит» секілді кең тараған). Кванттық теория және кванттық технологиялар жетістіктері логикалық және термодинамикалық қайтымды есептеулер теориясының құрылуына түрткі болды. А. Тьюринг және Дж. Фон Нейманның жұмыстарынан кейін қайтымды есептеулердің тұжырымы қалайда жаңа ойлау үрдісі – «кванттық мидың» құрылуына әкелетіні анық болды.

«Қарапайым» компьютерлерге арналған бағдарламалар компьютердің бұрынғы тарихын тастап қоятын амалдар жасайды, яғни машинаны тікелей бұрынғы жағдайына қатысты мәліметтері белгісіз жағдайда қалдырып қояды. Мұндай амалдар, мысалға, деректерді өшіру немесе «ескі» деректің үстіне жазу амалдарын жатқызуға болады.

Анықтама.

(1)( [Кван­то­вый,1999,с.18] бойынша) Логикалық қайтымсыз құрылғы деп шығысындағы сигнал арқылы кірісіндегі сигналын анықтау мүмкін болмайтын құрылғыны айтамыз.

(2)( [Кван­то­вый,1999,с.34] бойынша) Компьютердің көшу функциясы(машинаның әрбір глобальды күйін келесі күйіне көрсететін дара функция(әрине, ондай болса)) бір мәнді кері функциясына ие болмаса оны логикалық қайтымсыз деп атаймыз.

Басқа жағдайларда жоғалып кетуі мүмкін ақпаратты сақтау арқылы логикалық қайтымсыз компьютерді әрқашан қайтымды қылуға болады. Мысалы, машинаға әрбір операцияны орындалған бойда жеткілікті түрде толық жазып отыратын (басынан бос) қосымша лента қосу арқылы машинаның бұрынғы күйін қазіргі күйі мен лентадағы соңғы жазбаның көмегімен анықтауға болушы еді. Бірақ та, Р.Ландауэр айтып кеткендей, бұл тек қажетсіз ақпаратты лақтырып тастау мәселесін тек кейінге шегереді,өйткені лента қайтадан қолданудан бұрын тазаруы тиіс. Осылайша, тоқтаған кезде барлық аралық нәтижелерді өшіріп, тек қана қажетті шығыс және бастапқы кіріс деректерін қалдырып отыратын қайтымды компьютерлер қажет. (Машина кіріс деректерін сақтай алуы қажет, әйтпесе, ол қайтымды бола алмайды және бұрынғыша кіріс деректері шығыс деректері деректері арқылы бір мәнді анықталмайтын есептеулер жүргізе береді).

Тьюрингтің универсалды машинасы деп Тьюрингтің кез келген машинасын алмастыра алатын Тьюринг машинасын айтады. Ол кірісінде бағдарлама мен кіріс деректерін қабылдап, осы бағдарламаның Тьюринг машинасының кіріс деректерімен есептекге болатын жауапты шығарады.

**Тьюрингтің әмбебап машинасы**

**Тьюрингтің әмбебап машинасының формальды анықтамасы.** Кез келген детерминді Тьюринг машинасының бағдарламасын жақша, сызықшалар, т.б. символдардан тұратын кейбір ақырлы алфавитті қолдана отырып жазуға болады; Бұл машиналық алфавитті Σ1 деп атайық. Онда Σ2 алфавиті және k кіріс лентарары бар машиналар класы үшін U Тьюрингтің универсалды машинасы болып алғашқы k лентасына кіріс мәндерін беріп, ал k+1 –ге кейбір М1 Тьюринг машинасының дұрыс жазылған кодын бергенде, U осы кіріс деректері бойынша М1 шығаратын жауапты беретін, немесе М1 осы деректерде тоқтамаса, шексіз ұзақ жұмыс істейтін k+1 кіріс лентары және \Sigma_1 \cup \Sigma_2 алфавитінен тұратын Тьюринг машинасын айтамыз.

**Тьюрингтің универсалды машинасы туралы теорема** ондай машина бар және өзге машиналар әрі кетсе квадраттық баяулықпен(яғни бастапқы машина t қадам жасаса, онда универсалды машина әрі кетсе ct² қадам жасайды ) әзірлейді деп тұжырымдайды. Бұл теореманың дәлелдемесі сындарлы. (мұндай машинаны құру қиын емес, тек мұқият баяндай білу керек) Теорема 1947 жылы Тьюригпен ұсынылып және дәлелденді.

**Бағдарламалар мен олардың ерекшеліктерін дәлелдеудің формальды тәсілдері**

БҚ анализінің дәстүрді тәсілдері бағдарламаның дұрыстығын дәлелдеумен байланысты(бағдарлама верификациясы). Бұл бағытқа П. Наура мен Р. Флоидтың жұмыстары бастау берген болды. Онда индуктивті немесе аралық тұжырымдамалық деп бағдарлама нүктесіне жазып қою идеясы түйінделген болатын және индуктивті тұжырымдар сәйкестігіне құрылған бағдарламаның жеке бөлігінің дұрыстығын дәлелдеу мүмкіндігі көрсетілді.(яғни оның алғышарты мен соңғы шартының сәфйкестігі).

Ч. Хоор верификация теориясына іргелі үлес қосты, ол бағдарламаның бөлігінің дұрыстығына кейбір логикалық жүйеде шығару арқылы жүргізу туралы идеялар айтқан болатын, ал Э. Дейкстра болса, бір уақытта алғышарт пен соңғы шарттың сәйкестігін және аяқталуын білдіретін әлсіз алғышарт түсінігін енгізді. Бағдарламаның дұрыстығын дәлелдеу тәсілдері белгілі бір дәрежеде бағдарламалауға пайдасын тигізді. Бұл тәсілдердің бағдарламаның орындалу жолы туралы ойлау тәсілдерін көрсететіні, бағдардамада түсініктеме қалдырудың ыңғайлы жүйесін беретіні және бағдарламалау тілдерінің құрылымдары мен семантикалары арасында өзара байланыс орнататыны айқындалды. Егер бағдарламаның әр түрлі қасиеттерін дәлелдеу немесе бағдарламаларды дәлелдеу теоремасы ретінде «бағдарлама анализі» терминінің кең түсінікті мағынасын қабылдасақ, онда тәсілдердің құндылығы анықталады. Жеке жағдайда бағдарламаның шығыс деректірінің мәндерінің өзгеруін, бағдарлама орындалуы кезінде операциялар санын, циклдың, қолданысқа түспеген бағдарлама бөлігінің бар болуын зерттеуге болады. Осылай, кейбір жеке жағдайларда верификация тәсілдері бағдарламалық кемшіліктерді анықтау үшін де қолданыла береді.

Кейбір логикалық жүйеде шығару түріндегі формалды дәлелдеме толық сенімді, алайда дәлелдемелер өте ұзақ және жиі түсініксіз болады.

**Бағдарламаның дұрыстығын дәлелдеу әдісмтемесі**

Қарастырылп отырған әдістеме төбелеріне жады операторлары, ал доғаларына оператордан операторға көшу сәйкестендірілген граф түріндегі алгоритмдердің дұрыстығын дәлелдеуге арналған. Бір төбені кіріс төбесі деп аламыз, оған бағдарламаның орындаулы басталатын оператор сәйкес келеді, ал шығыс төбелері бірнеше болуы мүмкін.Кіріс және шығыс төбелері сәйкесінше кіріс және шығыс тілдерімен белгіленген деп есептейміз.

**Алгоритмнің дұрыстығын дәлелдеу дегеніміз төмендегідей тұжырымды дәлелдеу деген сөз: «Егер кіріс деректері кіріс шартын қанағаттандырса, онда алгоритм ақырлы қадамнан кейін жұмысын тоқтатып, шығыс деректері талап етілген шығыс шартына сейкес келеді».**

Тәжірибе жүзінде, мұндай тұжырымды 2-ге бөледі:

«Егер кіріс деректері кіріс шартын қанағаттандырса, онда алгоритм ақырлы қадамнан кейін жұмысын тоқтатып, шығыс деректері талап етілген шығыс шартына сейкес келеді».

«Егер кіріс деректері кіріс шартын қанағаттандырса, онда алгоритм ақырлы қадамдардан кейін жұмысын аяқтайды».

(1) тұжырым ғана дәлелденген алгоритм жартылай дұрыс немесе жартылай түзетілген деп аталады. Егер де (1) және (2) бар болса, онда алгоритм дұрыс немесе түзелген деп аталады.

(2) тұжырымды дәлелдеу жеңе алмайтын қиындықтар тудырғанда, (1) ғана дәлелдеумен шектелетіндігін айта кетейік. Мысалға, ұқсастық облысы белгісіз болатын итерациялық алгоритмдер сондай. Ондай кезде, егер алгоритм құптауға болатын уақытта жұмысын аяқтаса, онда жауаптың дұрыстығы кепілденеді.

**Жартылай дұрыстықты дәлелдеуде тоқталайық.**

Әдістеме келесіде қорытылады:

Есептеу қадамын бақылау үшін бақылау доғалары таңдалады. Алгоритмнің граф-сызбасында барлық циклдар бөлінген болуы үшін бақылау доғаларына міндетті түрде кіріс және барлық шығыс доғалары, сонымен қатар, басқа доғалардың да кейбір саны жатқызылады.

Әрбір бақылау доғасы үшін қарастылып отырған доға арқылы әрбір өткен сайын жадының ішіндегісі шамамен қанағаттандыратындай индуктивті шарт анықталады. Барлық бақылау доғаларын кейін бақылау нүктелері деп атайтын боламыз деп есептеп, және оларға сәйкес индуктивьі тұжырымдарды нөмірлейміз.

Блок-сызбада i, j ден тұратын басқа бақылау нүктелерін соқпай өтетін жолы бар әрбір i , j бақылау нүктелерінің жұбы үшін, барлық жолдар таңдалады және әрбір таңдалған жолға мынадай тұжырым дәлелденеді(индуктивті қадам): «Егер і нүктесі арқылы кезекті өту кезінде Рі индуктивті жорамалы орындалса және қарастылып отырған жол жүргізілсе, онда j нүктесіне жеткенде Pj шарты орындалады».

Егер барлық индуктивті қадамдар дәлелденсе, онда математикалық индукция принципін қолдана отырып, алгоритмнің жартылай дұрыстығын тұжырымдауға болады. Толық дұрыстығын дәлелдеу үшін бағдарламаның ақырлы қадамдардан кейін тоқтайтынын дәлелдеу қалады.

**Есептелу және шешімі болу.** n сөзден тұратын А алфавитіндегі реттелген жинақ А-ның үстіндегі n-орынды жинақ деп аталады. Оны (*A*\*)*n* деп белгілейміз.

Кез келген (*A*\*)*n* жиының R жиыншасы n-орынды сөздің қатынас деп аталады.

*f*: (*A*\*)*n* → *A*\* кез келген, мүмкін жеке көрсетілуі n-орынды сөздің функциясы деп аталады. *f*  функциясының анықталу облысы Def (*f* ) арқылы белгіленеді.

Т бағдарламасының жұмысының Х псевдосөзіндегі нәтижесі *T* (*x*) псевдосөзі деп аталады, ол бағдалама тоқтаған кезде лентада пайда болады; егер бағдарлама ақырсыз жұмыс жасаса, нәтиже анықталмаған.

Кез келген Х псевдосөзімен жұмыс кезінде бастауын n-ші Х псевдосөзіненің сол жағында орналасқан пробелден солға қарай жылжытпайтын бағдарламаны n-бағдарлама деп атаймыз.

Егер

*X\** *un \** *un*–1 *\** …*\*u*1 *\**↓ , где (*u*1, *u*2, …, *un*) ∈ *R*.

түрдегі барлық псевдосөдерде анық тоқтайтын Т n-программасы бар болса, R cөздік n-орынды қатынасы жартылай шешімді деп аталады.

Егер R және R(мұндағы R және R ретінде (*A*\*)*n* \ *R* түсініледі) жартылай шешімді болса, R сөздік n-орынды қатынасы шешімді деп аталады.

Егер

*T*(\**u*1\**u*2\*...\**un***\***↓ ) = \**u*1\**u*2\*...\**u*n\**v***\***↓ ,

мұндағы *u*1, *u*2, ..., *un* ∈ Def (*f* ) және *v* = *f* (*u*1, *u*2, ..., *un*) болатын Т бағдарламасы бар болса, *f*: (*A\**)*n* → A\* сөздік n-орынды функциясы Тьюринг бойынша есептелінеді деп аталады, кері жағдайда нәтиже анықталмаған.

Тьюринг бойынша есептелетін функцияларды жартылай есептеленуші, ал рұқсат етілген анықталу обылсымен жартылай есептелінушілерді – есептелінуші деп атауға болушы еді, алайда бұл қалыптасқан дәстүрге қарсы.

Сандық функцияларды есептеу

Сандық функциялардың мәнін тьюринг бағдарламаларымен есептеу үшін аргументтер лентасында және функция мәндерін кодтау тәсілін таңдау қажет. Біз N-дегі *N n* функциясынан қарастырамыз, мұндағы N – 0-ді қосқандағы натурал сандар жиыны, ал *n* ≥ 1.

Функция және оның аргументтерінің мәнін бинарлы, унарлы немесе кез келген бір кодта жазамыз, бұл үшін бізге сәйкес *A* = {1}, *A* = {0, 1} т.б. алфавиті қажет болады. Аргументтердің мәні есептелуден бұрын лентада

\**x n*\**xn*–1\*...\**x*1**\***↓ ,

мұндағы *xi* – *i*-ші аргументтің коды (*i* = 1, 2, ..., *n*).

псевдосөзі түрінде көрсетілуі тиіс.

Есептелуден соң лента ішіндегісі мына түрде болады:

\**x n*\**xn*–1\*...\**x*1\**y***\***↓ ,

мұндағы *y* – берілген аргументтің мәнінде функция мәнінің коды.

**ҚОРЫТЫНДЫ:**

Кванттық теория және кванттық технологиялар жетістіктері логикалық және термодинамикалық қайтымды есептеулер теориясының құрылуына түрткі болдыЛогикалық қайтымсыз құрылғы деп шығысындағы сигнал арқылы кірісіндегі сигналын анықтау мүмкін болмайтын құрылғыны айтамыз.Компьютердің көшу функциясы(машинаның әрбір глобальды күйін келесі күйіне көрсететін дара функция(әрине, ондай болса)) бір мәнді кері функциясына ие болмаса оны логикалық қайтымсыз деп атаймыз.Тьюрингтің универсалды машинасы деп Тьюрингтің кез келген машинасын алмастыра алатын Тьюринг машинасын айтады. Ол кірісінде бағдарлама мен кіріс деректерін қабылдап, осы бағдарламаның Тьюринг машинасының кіріс деректерімен есептекге болатын жауапты шығарады.Алгоритмнің дұрыстығын дәлелдеу дегеніміз төмендегідей тұжырымды дәлелдеу деген сөз: «Егер кіріс деректері кіріс шартын қанағаттандырса, онда алгоритм ақырлы қадамнан кейін жұмысын тоқтатып, шығыс деректері талап етілген шығыс шартына сейкес келеді».