

Нека F{f1 (x1…xn)….fk(x1…xn)} е множество от двоични функции

Def: Под формула на F ще разбираме

1. Всички ф-ии от F

2. Всички изрази от вида fi(f1,f2; fni), където fi ∈ F, а fi са или формули над F, или буква или променлива.

Напр: Ако F = {f1(x1), f2(x1,x2)}, то f1(x2), f2(x1,x2), f2(f1(x1)), f2(f1(x1), f1(x2)), f2(f1(x1), f1(f2(x1, x2)) са формули над F

Def: Ако функцията Р може да се представи с формула над F, казваме, че е суперпозиция над F.

Def: Множества от всички суперпозиции над F ще наричаме ЗАТВОРЕНА ОБВИВКА на F и ще означаваме с [F], т.е. [F] се състои от всички ф-ии, които могат да се реализират с ф-ли над F.

Def: Елементарна конюкция е израз от вида x1, x2…xk, k=1, n, ∀n ∈ N без повтарящи се множители

Пример: x1 x2 x3 x1 – не ;

x1 x2 x3 - да

**Самоидвоинствени: множеството S**

Нека f(x1, x2… xn)

Def1 f\* е двоинствена на f, ако f\* = ̅f( ̅x1, ̅x2… ̅xn)

Def2 f е самодвоинствена, когато f = f\*

**Монотонни: множеството М, което е затворено**

Ако λ= <а1, а2, an> и β = <b1, b2, bn> са две числови последователности, казваме че λ предхожда β или λ< β, ако ai ≤ bi за i = 1, 2 … n

**Линейни двоични функции:** L е затворено  
Функцията f(x1,x2…xn) е линейна, когато ПЖ е линеен.  
Функцията f(x1, x2, xn) е **линейна**, когато ПЖ е линеен, т.е. f(x1... xn) = a1x1+a2x2+anxn+a0, където ai=0 или 1.

**Теорема 3.1:**

С-затворено, ако:

1. xi ∈ C (идентитетите са в това множество)

2. Ако f, g…gn ∈ C => f(g1…gn) ∈ C

**Теорема 3.2:**

T0, T1 – затворени

1. X1 ∈ 0

2. Нека f, g1, gn ∈ T0

f(g1(0…0)…, gn(0…0))=0 => ∈ T0

**Теорема 3.3**

Нека

f0 ∉ T0 => f0(0…0) = 1

f0(1….1) = a

f1 ∉ T1 = > f1(1…1) = 0

f1(0…0)=b

g0(x1) = f0(x1….x1)

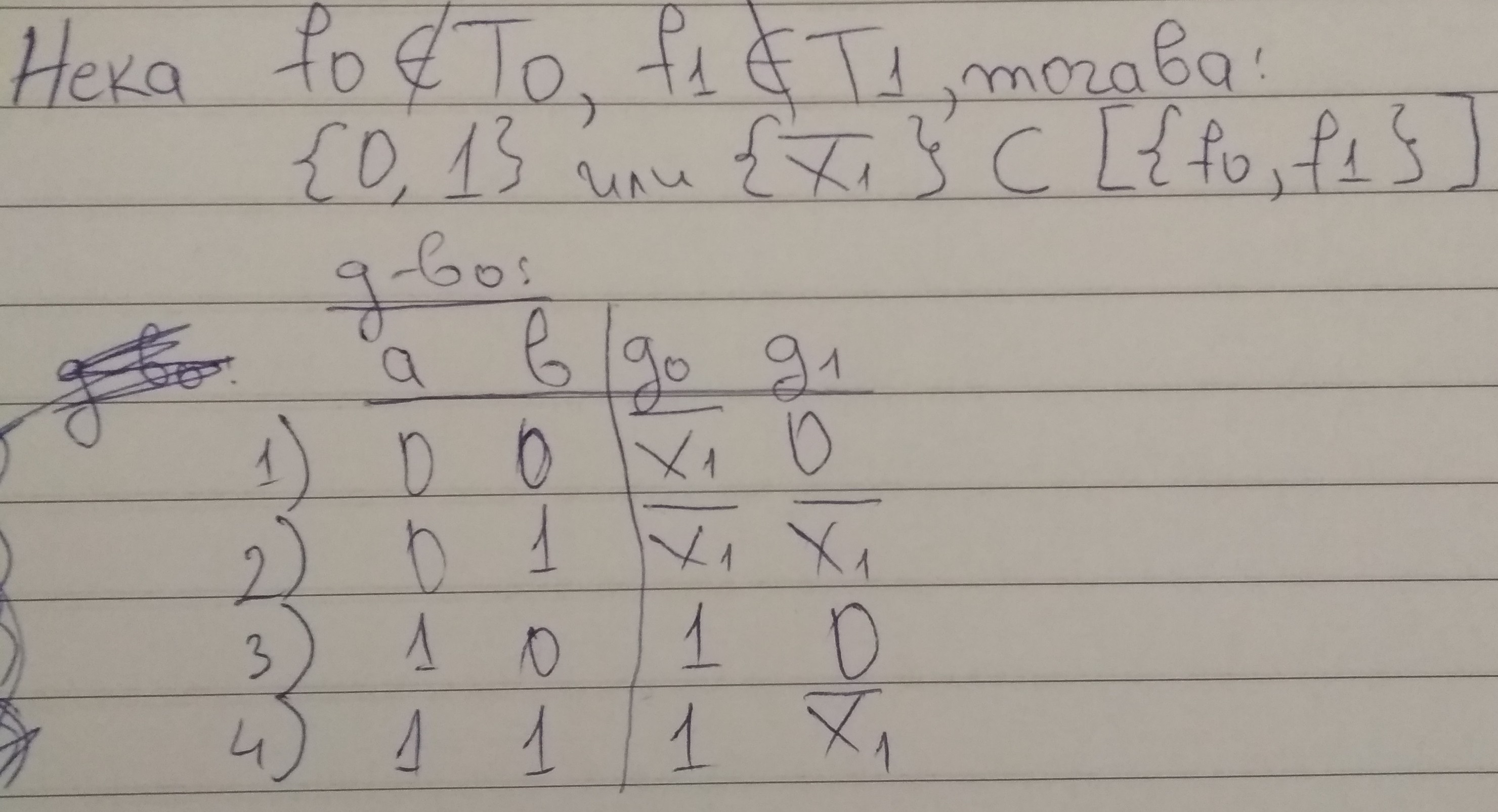
g0(0) = 1

g0(1) = a

g1(x1) = f1(x1…x1)

g1(0) = b

g1(1) = 0

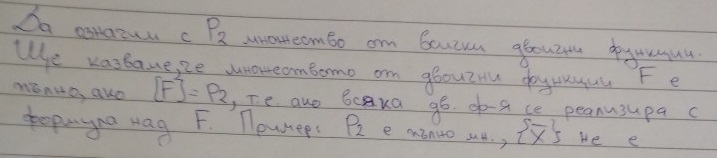
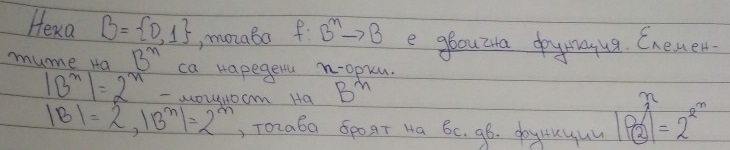


Теорема 4.1

Нека f0 ∉ T0, f1 ∉ T1, fs ∉ S {0;1} ⊂ [{f0, f1, fs}] => Множеството S е затворено

Д-во: Нека съгласно теорема 3.3 сме реализирали { ̅x1}

fs ∉ S => съществува <a1.. an> : fs (a1..an) = fs (an)



**Полином на Жегалкин:**

Израз от вида E1 + E2 + … + Ek, k=1-n, без повтарящи се събираеми, където Ei = {1 елем. конюкция}  
За две променливи: ax1x2+b1x1+b2x2+c  
За три променливи: ax1x2x3+b1x2x3+b2x1x3+b3x1x2+c1x1+c2x2+c3x3+d

**Теорема на Жегалкин**: Всяка двойчна функция се представя с точно един полином на Жегалкин. Т.е. всяка двойчна двойчна функция на n-променливи се представя чрез ПЖ. Всички ПЖ са 2^(2^n)

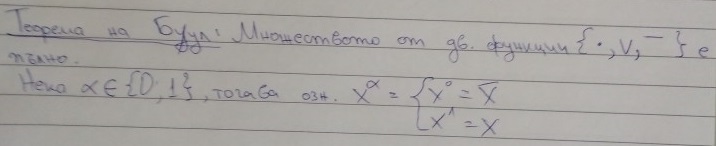
[F] - **Обвивка на F** – Множество на всички функции на F

Едно множество е **пълно**, само ако неговата обвивка = P2

Нека F={f1, f2…fn} е пълно. Можеството G = {g1…gk} е пълно, тогава и само тогава, когато за всяка двоична функция fi ∈ F => fi ∈ G[G], т.е всяка двоична функция от F Може да се представи чрез формула над G.

**Теорема на Бул:** Множеството от двоични ф-ии F{ ·,⌵, ─ } – пълно

Нека λ ∈ {0;1}, товага означ. xλ={x0 = x ̅ ; x1=x }



Функцията f(x1, x2, xn) е **линейна**, когато ПЖ е линеен, т.е. f(x1... xn) = a1x1+a2x2+anxn+a0, където ai=0 или 1.

Множеството **L** е затворено , т.е. съвпада със своята обвивка

Множеството **М** е затворено.

**Теорема на Пост**: Едно множество функции на F е пълно, само когато F ⊄ T0, T1, M, S, L

Функцията f е **Шеферова**, само когато f не принадлежи на T0 ∪ T1 ∪ S  
Функцията f е **Шеферова,** ако сама образува пълно множество.

**ГРАМАТИКА**

**Азбука**: Крайно множество от символи.

Азбука V – всяко непразно крайно множество от символи.  
Дума над азбуката V е всяка крайна последователност от символи.

**Дължина на дума** **| λ|** = броя букви.

**Празна дума** [**Ɛ**](https://en.wikipedia.org/wiki/Latin_epsilon) е дума, която няма буква. Дължината й е 0.

**V\*** е множеството на всички думи над V. Няма край.

**Формална граматика**:Формален език (множество от думи) над азбуката V е **L** (language)**.**

**Г** = наредени множества {V, W, S, p}

**V** – азбука от терминални символи. (Изходна)

**W**- нетерминални символи (Генериращи)

V ∪ W = ∅

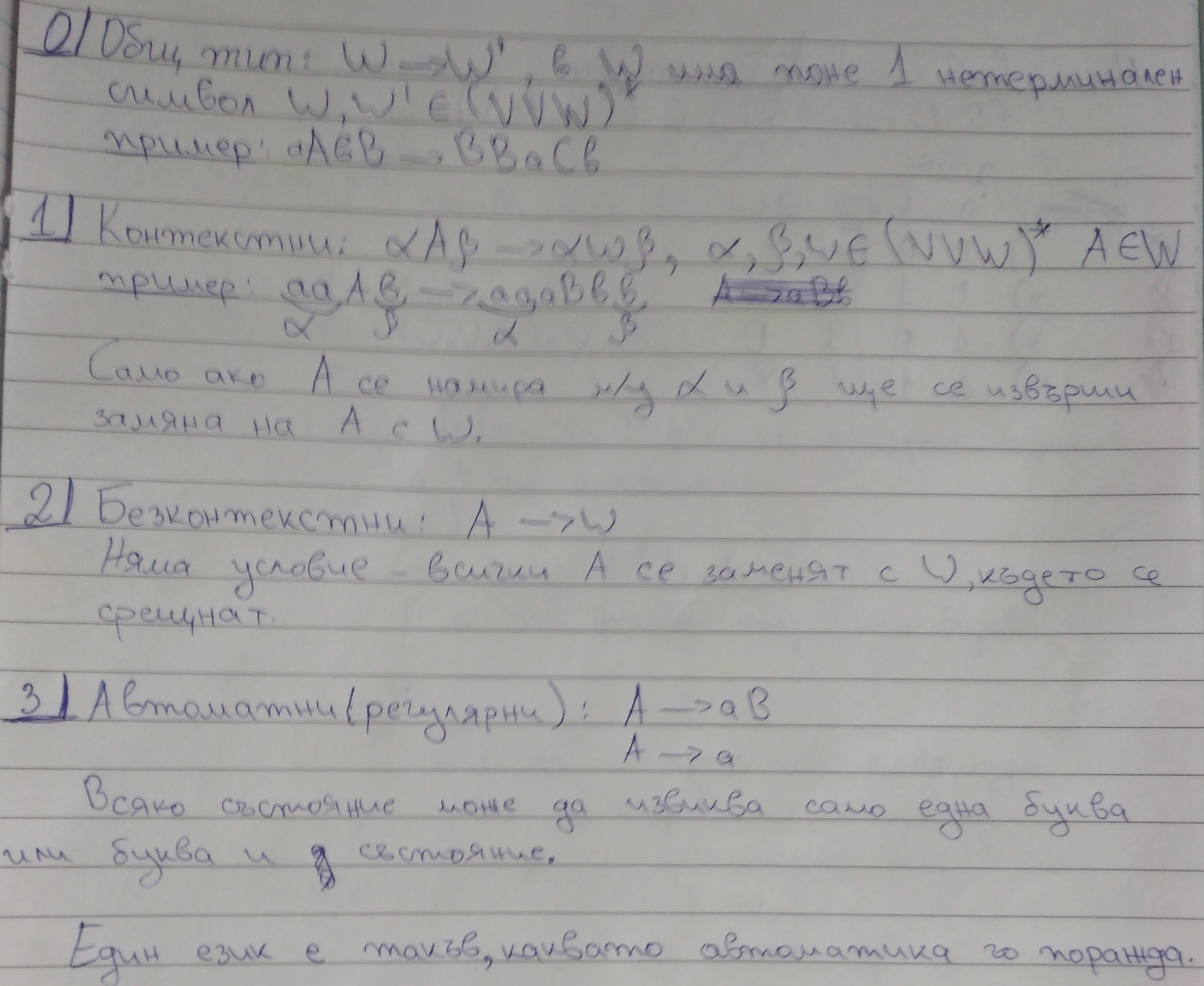
**S** – символ от W – начално състояние.

**Р** – крайно множество от правила.

**Автоматна** – ако прехода от едно към друго състояние е единствен.

**Неавтоматна** – ако позволява повече преходи между състоянията

**Йерархия на Чомски (Chomsky):**



В граматики от вид 1,2 и 3 може да има правило S-> Ɛ, ако S не се среща отляво в правилата.

**Теорема 10.1**: Нека L‘ и L’’ са автоматни, тогава L’∪L’’ също е автоматна.

Д-во: L’ авт. => съществува АГ = <V’, W’, S’, p’> : L (Г‘) = L’

L’’ –авт. => <V’’, W’’, S’’, P’’> : L (Г’‘) = L’’

Конструираме АГ <V’ ∪ V’’, W’ ∪ W’’ ∪ (S’), S, P >

P= p’ \{S’ -> ε } ∪ P’’ \ { s ‘’ -> ε } ∪ {S -> α ⇔ S’ - > α ∈ P’ или S’’-> α P’’}

1. Нека ω ∈ L’ ∪ L’’ => … => ω ∈ L(Г)

2. Нека ω ∈ L(Г) => ... => ω ∈ L’ ∪ L’’

=> L’ ∪ L’’= L(Г)

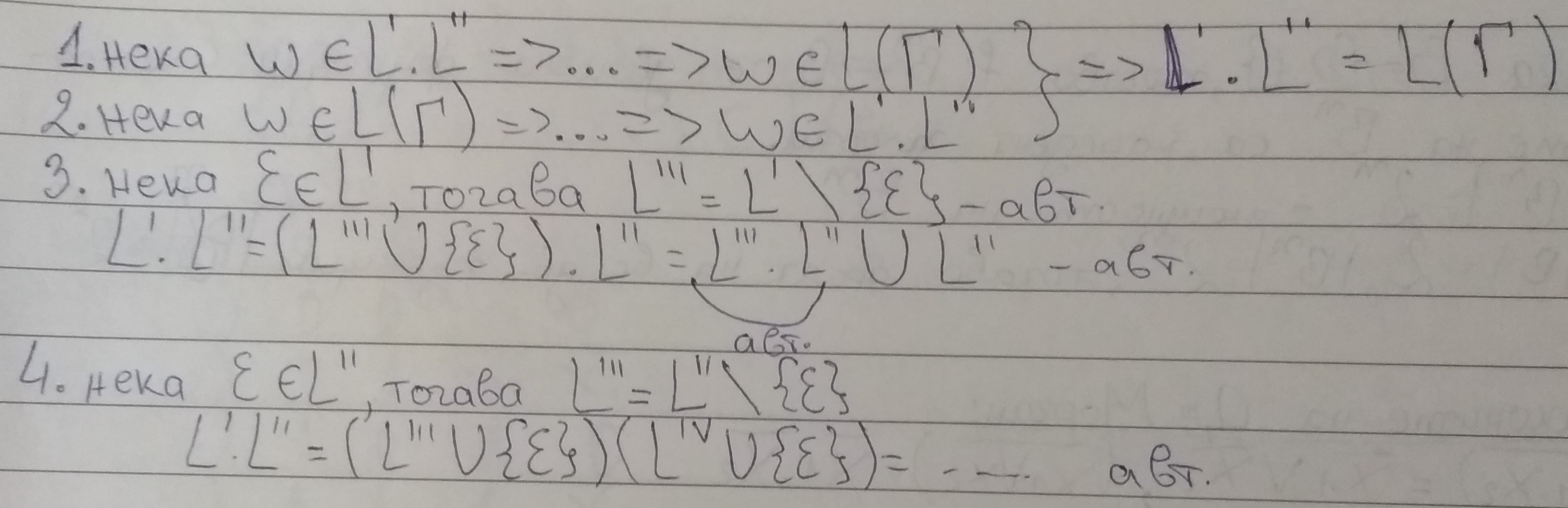
**Теорема 10.2:**

L’.L’’ – автоматна граматика

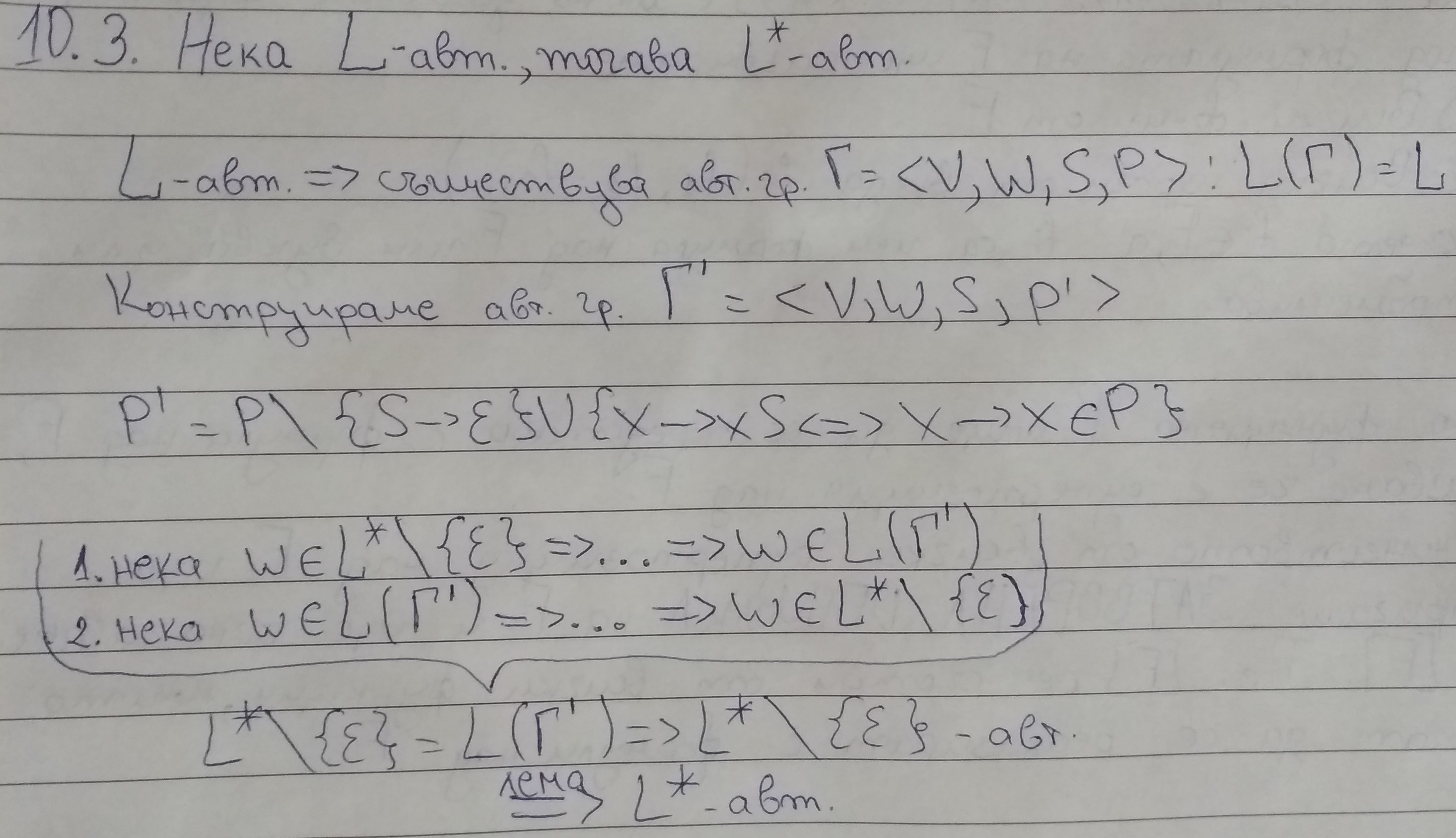
Нека ε ∉ L’ ∪ L’’

Конструираме авт. гр. Г = <V’ ∪ V’’, W’ ∪ W’’, S’, p>

P= p’’ ∪ p’ \ {x’ → x ∉ p’} ∪ {x’ → xS’’ ⇔ x’ → x ∈ p’}



**Теорема 10.3:**



**Теорема 10.4:**

Всеки краен език е автоматен.

Д-во: Очевидно всеки език от 1 дума е автоматен, томага крайният език представлява обединение на краен брой такива езици и съгласно Th. 10.1 е автоматен

ДКА – Частен случай на НДКА.

ДКА и НДКА разпознават еднакъв брой езици.

???????????????????????

Нека L се разпознава от ДКА, тогава съществува const  
n, n=n(A) (n- брой вътрешни състояния)  
L∈α, |α|>=n, α=uv ω, |u|<n, |V|>0

uv ω∈L, i=0,1,2,…

???????????????????????

**Теорема1:**

Нека L се разпознава от ДКА, тогава L е автоматен.

Д-во: Нека L се разпознава от ДКА А = <K, V, δ, qo, F>

Конструираме АГ: Г=<V, K, qo, P >

P = (дълго): qi → xqj ⇔ δ(qi, x) = qj

(късо): qi → x ⇔ δ(qi, x) ∈ F

1. Нека ω ∈ L(r) => … => ω ∈ L \ {ε}

2. Нека ω ∈ L \ {ε} => ... => ω ∈ L(r)

L(r) = L \ {ε}

=> L \ {ε} – авт. => (лема) L – авт. □

**Теорема2:**

Нека L е автоматен, тогава той се разпознава от НДКА.

Д-во: Нека L се поражда от АГ: Г=<V, W, S, P>

Конструираме НДКА А=<W ∪ {F}, V, δ, S, {F} ∪ {S}, ако ε ∈ L >

\*F∉W

δ : Y ∈ δ(X,x) ⇔ x → xY

F∈ δ(X,x) ⇔ x → x

1. нека ω ∈ L => ... => ω∈Т(А)

2. нека ω∈Т(А) => ... => ω ∈ L

=> L=T(A) □

**Теорема Rabin и Scott:**

За всеки НДКА съществува еквивалентен ДКА.

Д-во: Нека L се разпознава от НДКА А = <K, V, δ , qo, F>

Конструираме ДКА А‘ = <K’, V, δ’, qo’, F’>

K’ = 2k – множеството от всички подмножества на К.

T[qo,1…qi] ↔ {q0,1….qi} ⊂ K (T[q1…qi] ↔ {q1….qi} ⊂ K)?? така съм записал аз

q0’ = t[q0] – начално

t[P1…Pn] ∈ F’ ⬄ { P1…Pn } ∩ F ≠ ∅

δ’ (t[r1….rm], x) = t[p1…pn] ⇔ {p1…pn} = δ (r1, x) ∪ δ (r2, x) ∪…∪ δ (rm, x)

1. Нека ω∈Т(А) => ... => ω∈Т(А‘)

2. Нека ω∈Т(А‘) => .... => ω∈Т(А)

=> Т(А) =Т(А‘)□