



УНИВЕРЗИТЕТ У КРАГУЈЕВЦУ  
ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА ЧАЧАК

Жељко Ј. Поповић

**РАЗВОЈ ЈЕЗИКА ЗА 2D  
ПАРАМЕТАРСКО ГЕОМЕТРИЈСКО  
МОДЕЛИРАЊЕ**

Докторска дисертација

Ментор: проф.др Алемпије Вељовић

Чачак, 2017

<b><i>I. Аутор</i></b>	
Име и презиме:	<b>мр Жељко Поповић</b>
Датум и место рођења:	<b>02.03.1957, Госпић</b>
Садашње запослење:	<b>Техникум Таурунум, Висока инжењерска школа струковних студија, Београд (Земун)</b>
<b><i>II. Докторска дисертација</i></b>	
Наслов: <b>РАЗВОЈ ЈЕЗИКА ЗА 2Д ПАРАМЕТАРСКО ГЕОМЕТРИЈСКО МОДЕЛИРАЊЕ</b>	
Број страница:	<b>144</b>
Број слика:	<b>29</b>
Број библиографских података:	<b>107</b>
Установа и место где је рад израђен:	<b>Факултет техничких наука Чачак</b>
Научна област (УДК):	<b>Техничка информатика</b>
Ментор:	<b>Проф. др Алемпије Вељовић</b>
<b><i>III. Оцена и одбрана</i></b>	
Датум пријаве теме: <b>14.10.2015.</b>	
Број одлуке и датум прихватања докторске дисертације: Број: <b>53-2165/14</b> Датум: <b>18.11.2015.</b>	
Комисија за оцену подобности теме и кандидата:	
<ol style="list-style-type: none"> <li><b>1. Др Алемпије Вељовић</b>, редовни професор, Факултет техничких наука Чачак Универзитета у Крагујевцу на наставним предметима <i>Менаџмент информациони системи</i> и <i>Менаџмент развоја</i>, научна област: <i>Техничко технолошке науке</i>, ужа научна област: <i>Менаџмент информациони системи</i>.</li> <li><b>2. Др Радомир Славковић</b>, редовни професор, Факултет техничких наука Чачак Универзитета у Крагујевцу, ужа научна област: <i>Производне технологије</i>.</li> <li><b>3. Др Иван Милићевић</b>, доцент, Факултет техничких наука Чачак Универзитета у Крагујевцу, ужа научна област: <i>Конструкционо машинство</i>.</li> <li><b>4. Др Марко Поповић</b>, доцент, Факултет техничких наука Чачак Универзитета у Крагујевцу, ужа научна област: <i>Конструкционо машинство</i>.</li> <li><b>5. Др Горан Путник</b>, visiting profesor Universidade do Minho, Braga, Portugal, Faculty Department of Production and Systems Engineering.</li> </ol>	
Комисија за оцену докторске дисертације:	
<ol style="list-style-type: none"> <li><b>1. Др Биљана Радуловић</b>, редовни професор, Технички факултет Михајло Пупин Зрењанин, ужа научна област: <i>Базе података и информациони системи</i></li> <li><b>2. Др Радомир Славковић</b>, редовни професор, Факултет техничких наука Чачак Универзитета у Крагујевцу, ужа научна област: <i>Производне технологије</i>.</li> <li><b>3. Др Иван Милићевић</b>, доцент, Факултет техничких наука Чачак Универзитета у Крагујевцу, ужа научна област: <i>Конструкционо машинство..</i></li> </ol>	

Комисија за одбрану докторске дисертације:

1. **Др Биљана Радуловић**, редовни професор, Технички факултет Михајло Пупин Зрењанин, ужа научна област: *Базе података и информациони системи*
2. **Др Радомир Славковић**, редовни професор, Факултет техничких наука Чачак Универзитета у Крагујевцу, ужа научна област: *Производне технологије*.
3. **Др Иван Милићевић**, доцент, Факултет техничких наука Чачак Универзитета у Крагујевцу, ужа научна област: *Конструкционо машинство*

Датум одбране дисертације:

## ЗАХВАЛНИЦА

*При изради овог рада имао сам несебичну помоћ ментора проф. др Алемпија Вељовића, редовног професора Факултета техничких наука Чачак Универзитета у Крагујевцу, коме се најсрданије захваљујем на значајним саветима и свесрдној подршци.*

*Свим срцем се захваљујем и својој породици, која ми је пружила безусловну подршку и охрабрење током писања докторске дисертације.*

*Чачак, 2017.*

*Мр Жељко Поповић*

# РАЗВОЈ ЈЕЗИКА ЗА 2D ПАРАМЕТАРСКО ГЕОМЕТРИЈСКО МОДЕЛИРАЊЕ

## Резиме

Глобализација савременог тржишта захтева производњу све сложенијих производа, са великим бројем варијанти пројектних решења, ради прилагођавања специфичним захтевима крајњег потрошача. Конкуренција на светском тржишту, посебно изражена у машиноградњи, ваздухопловној индустрији, индустрији моторних возила, електронској и сличним индустријама, захтева стално повећање квалитета производа, смањење цена, смањење времена изласка производа на тржиште и флексибилну производњу. Посматрано са техничко-технолошког аспекта, услов опстанка на тржишту је стално увођење нових технологија, као што су: програмабилна и флексибилна аутоматизација, компјутерски интегрисана производња и нови концепти, као што су агилни производни системи, интелигентни производни системи и слично. Параметарско пројектовање има могућност контроле модела параметарски дефинисаним „водећим” димензијама, као и увођења недимензионих геометријских и других ограничења. Параметар као информација која у потпуности одређује модел протеже се кроз све нивое система, а савремени *CAD/CAM* системи имају двосмерну асоцијативност (измена параметра на неком нивоу – моделу изазива промену на свим осталим нивоима – моделима). Променом вредности параметара који дефинишу геометрију, аутоматски се мења и путања алата, што значи да се на овај начин може остварити брза измена *CNC* програма – за делове, који су геометријски и технолошки слични, или – за нове варијанте производа. На основу референтног модела представника групе реализују се експлицитни модели појединих чланове групе променом вредности параметара модела.

**Кључне речи:** *2D геометријско моделирање, параметарско моделирање, програмски језик, пројектантска база података, препроцесор.*

## Abstract

The globalization of today's market requires the production of increasingly more complex products, with a large number of variants of design solutions, in order to accommodate the specific requirements of the end customer. Competition in the global market, especially strongly expressed in mechanical engineering, aviation industry, industry of motor vehicles, electronics and related industries requires steadily increasing product quality, reducing prices, reducing the time – to market and flexible production. Being seen from the point of technical and technological aspect, condition of survival on the market is: constantly introducing new technologies as programmable and flexible automatization, computer integrated manufacturing and new concepts such as agile production systems, intelligent production systems and similarly. Parametric designing has the ability to control the model parametrically defined as "major" dimensions, as well as introducing no-dimensional geometric and other restrictions. Parameter as information which fully defines model, is extending through all levels of the system and advanced *CAD/CAM* systems have bidirectional associativity (changing parameter at some level causes a change in the model for all other levels – models). By changing the values of parameters that define the geometry, automatically is, changing the tool path, which means that in this way is possible to achieve rapid shift of *CNC* programs, for the parts that are geometry and technologically similar or for new variants of products. Based on the reference model as representative of the group, we can see that an explicit models of individual members of the group are realising by changing the values parameters of model.

**Keywords:** *2D Geometric Modeling, Parametric Modeling, The Programming language, Project databases, Preprocessor.*

# САДРЖАЈ

ЗАХВАЛНИЦА .....	II
РЕЗИМЕ .....	II
АВСТРАСТ .....	III
САДРЖАЈ .....	IV
ПРЕГЛЕД СЛИКА .....	VI
1. УВОДНА РАЗМАТРАЊА.....	1
1.1. Предмет и циљ истраживања.....	1
1.2. Основне хипотезе.....	2
1.3. Преглед стања у подручју истраживања.....	3
1.4. Методе истраживања .....	6
1.5. Значај истраживања .....	6
2. ТРЕНД РАЗВОЈА И УТИЦАЈ НА ИНДУСТРИЈУ .....	8
2.1 Место и улога рачунара у процесу конструисања .....	8
2.1.1. Методе геометријског моделирања [33] .....	10
2.1.2. Процес конструисања (структура и окружење) [33].....	11
2.1.3. Процесори, препроцесори и постпроцесори [33].....	12
2.1.4. Нумерички управљане машине алатке – <i>NC</i> програмирање [33] .....	13
2.2. Уопште о <i>CAD</i> системима и параметарском моделирању.....	13
2.2.1. <i>2D CAD</i> системи .....	15
2.2.2. Параметарски модел .....	15
3. ЈЕЗИК ЗА 2D ПАРАМЕТАРСКО ГЕОМЕТРИЈСКО МОДЕЛИРАЊЕ .....	23
3.1. Општа структура.....	23
3.2. Елементи у структури примарног (геометријског) описа ентитета .....	27
3.3. Дефиниције примарног (геометријског) описа ентитета .....	34
4. СТРУКТУРА ЈЕЗИКА ЗА ПАРАМЕТАРСКО ГЕОМЕТРИЈСКО МОДЕЛИРАЊЕ <i>2/3D</i> .....	39
4.1. Језик за <i>2D</i> параметарско геометријско моделирање.....	47
5. ДЕФИНИЦИЈЕ КОМАНДИ (КОМАНДНИХ ЛИНИЈА) ЗА МОДЕЛИРАЊЕ ЕНТИТЕТА .....	53
5.1. Оријентисана дуж .....	53
5.2. Кружница .....	58
5.3. Кружни лук.....	61
5.4. Специфични примери дефинисања основних ентитета .....	63
5.5. Ентитети који су погодни за интерактивно моделирање са једним подтипом релационих исказа .....	66
5.6. Дефиниције за моделирање дужи паралелних <i>x</i> или <i>y</i> оси .....	68
5.7. Многоугао.....	69
5.8. Правоугаоник .....	69
5.9. Крива линија.....	70
6. ПРИМЕНА ЈЕЗИКА ЗА <i>2D</i> ПАРАМЕТАРСКО ГЕОМЕТРИЈСКО МОДЕЛИРАЊЕ.....	71
6.1. Примери описаних форми машинских елемената.....	71
6.2. Фиктивни ентитети.....	72
6.3. Примена релационих исказа на једном фабричком типском елементу (стуб за алат)....	75
6.4. Примена релационих исказа и фиктивних ентитета на једном фабричком типском елементу (стуб за алат).....	76
6.5. Програмска реализација примарног описа кривих линија.....	77

6.6. Кратки опис језика за $2D$ параметарско геометријско моделирање и препроцесора за његово коришћење .....	78
6.7. Моделирање (друга фаза) за издвајање једног од два скупа тачака (једне колекције геометријских података) када се користе релациони искази.....	82
7. ПРОЈЕКТАНТСКА БАЗА ПОДАТАКА КАО ПОДРШКА ЈЕЗИКУ ЗА $2D$ ПАРАМЕТАРСКО ГЕОМЕТРИЈСКО МОДЕЛИРАЊЕ .....	85
8. ДИСКУСИЈА О РЕЗУЛТАТИМА .....	100
9. ДАЉА ИСТРАЖИВАЊА.....	107
10. ЗАКЉУЧНЕ НАПОМЕНЕ .....	112
11. ЛИТЕРАТУРА .....	115
12. ПРИЛОЗИ .....	120



## ПРЕГЛЕД СЛИКА

- Слика 1: Етапе у развоју и коришћењу производа од идеје до рециклаже (уништења)
- Слика 2: Примена рачунара у процесу конструисања и производњи
- Слика 3: Структура *CAD*-а и веза са *CAM*-ом
- Слика 4: Околина *CAD* и потреба за препроцесорима и постпроцесорима
- Слика 5: Пример параметарског објекта
- Слика 6: Модел троугла за пример ограничења
- Слика 7: Пример конструкције помоћу граничне геометрије укључујући конструкционе линије
- Слика 8: Дуж тангентно на две кружнице I
- Слика 9: Дуж тангентно на две кружнице II
- Слика 10: Дуж тангентно на две кружнице III
- Слика 11: Пример моделирања кружног лука помоћу фиктивне кружнице
- Слика 12: Коришћење фиктивних ентитета код моделирања машинског елемента чеп за алат
- Слика 13: Стуб за алат
- Слика 14: Модул за генерисање малих *LISP* програма из командне линије за опис ентитета
- Слика 15: Пример садржаја пројектантске базе података
- Слика 16: Део поглавља о опругама у пројектантској бази података
- Слика 17: Подаци за моделирање у табели
- Слика 18: Апроксимација дијаграма једначином
- Слика 19: Презентовање моделираног пробојца у пројектантској бази података
- Слика 20: Пример преласка на поглавље у пројектантској бази података
- Слика 21: Форма са слике 7. са ознакама ентитета
- Слика 22: Пример ентитета дуж тангентно на кружницу
- Слика 23: Пример ентитета кружница тангентно на две дужи
- Слика 24: Пример ентитета кружница са датом тачком центра тангентно на другу кружницу
- Слика 25: Селекција помоћу изломљене линије
- Слика 26: Селекција помоћу три избора
- Слика 27: Избор константе
- Слика 28: Избор формуле
- Слика 29: Избор параметара из табеле

# 1. УВОДНА РАЗМАТРАЊА

## 1.1. Предмет и циљ истраживања

Систем за пројектовање и одговарајући програмски језик припадају научно-техничкој области примене рачунара за област машиноградње и обрадних технологија које јој припадају, у ужем смислу системима *CAD (Computer Aided Design – Компјутерски подржано пројектовање)*.

*2D* параметарско геометријско моделирање омогућава креирање дводимензионалног објекта. Пројектовање се основа за *2D* параметарско геометријско моделирање, тј. његова теоријска поставка.

Следи формулација конкретног програмског језика (језика за *2D* параметарско геометријско моделирање) који обухвата реализацију којом се омогућава стварање језика за параметарско моделирање. Очекује се да крајњи резултат буде опште примењив језик за *2D* параметарско геометријско моделирање. Овај језик примењују инжењери и техничари машинске струке производног смера, јер осим за геометријско моделирање, може да се користи и код пројектовања технологије на *CNC* машинама. Производња препроцесора би се извршила на бази најпознатијег светског система за цртање *AutoCAD*.

Активности ће се изводити на три најзначајнија нивоа:

### 1. Израда опште структуре језика.

Овде се подразумева израда структуре компјутерског програмског језика специјалне намене, али без лексичке, семантичке и синтаксичке структуре. Структура треба да је таква да врхунски стручњаци могу на бази ове опште структуре развити разне варијанте техничког извођења језика за *2D* параметарско геометријско моделирање.

### 2. Лексичка, семантичка и синтаксичка структура језика за *2D* моделирање.

Овде је потребно израдити детаљну лексичку, семантичку и синтаксичку структуру. Структура треба да је таква да омогућава угодан рад корисника овог компјутерског програмског језика. Овде се већ даје једна варијанта техничког извођења на бази опште структуре. Ова структура, као и структура дата под 1, треба да су такве да се може језик практично употребљавати коришћењем препроцесора.

### 3. Израда препроцесора за коришћење језика за *2D* параметарско геометријско моделирање.

Препроцесор је израђен на бази једног од најпознатијих пакета *AutoCAD*. Препроцесор је израђен у програмском језику *AutoLISP* зато што израда процесора захтева изузетно велика улагања.

У процесу израде дисертације користиће се следеће основне претпоставке:

- преглед тренутног стања, као и анализа постојећих система за геометријско моделирање,
- теоријска поставка концепта новог језика за *2D* параметарско геометријско моделирање,
- поседовање одговарајућих софтверских алата за проверу функционисања језика за *2D* параметарско геометријско моделирање,
- израда програмског пакета који омогућава функционисање језика за *2D* моделирање.

Овим језиком може се постићи једноставно и директно моделирање форме техничких објеката и других графикана или геометрије код *CNC* машина за обраду, уз могуће моделирање на разне начине, и то лако и без крутих правила. Такође, овим се желело постићи моделирање што већег броја могућих форми, и то на исти начин, са конкретним димензијама (константама), општим димензијама (варијаблама) и комбинацијом конкретних и општих димензија.

Анализом досадашњих истраживања поступака моделирања, када на основу једног дела описа ентитета постоји више решења за ентитет који се моделира, показује се да овај проблем није решен на погодан начин.

Општи циљ је да се добије основа за пројектовање *2D CAD* или *CAD/CAM* система. Посебни циљ је да се формулише један компјутерски програмски језик, односно језик за *2D* параметарско геометријско моделирање. Задатак нам је да покажемо како се на једноставан начин могу описати разне форме техничких објеката, односно да такав програмски језик може да се користи код описа геометрије за *CNC* машине за обраду метала, дрвета итд.

Коришћењем новог језика за геометријско моделирање (*2D*) знатно се олакшава опис објеката увођењем нових елемената код описа основних ентитета. Код увођења нових елемената у виду исказа који описују ентитет, а који су блиски корисницима, значајно се повећава број дефиниција основних ентитета које се користе код описа неке форме.

## 1.2. Основне хипотезе

Основна хипотеза докторске дисертације је да се у једном делу решења за дефиниције ентитета може омогућити знатно повећање перформанси у односу на постојеће системе и језике за *2D* параметарско геометријско моделирање.

Друга хипотеза је да се могу истим језиком моделирати машински елементи са општим димензијама (варијаблама), конкретним димензијама и комбинацијом општих и конкретних димензија.

Трећа хипотеза је да се може описати и геометрија код *CNC* машина (*2D*).

### 1.3. Преглед стања у подручју истраживања

Први графички систем развијен је 1950. године за потребе Америчког ратног ваздухопловства. То је био систем против ваздушне одбране (полуаутоматизовано окружење). Систем је развијен на *Масачусетском институту* код *Линколн* технолошке лабораторије, који је такође укључивао употребу *CRT* монитора да би се приказале информације добијене од радара, али и многе друге. Као почетни период развоја *CAD/CAM* система узима се 1960. година, и то у авионској и аутомобилској индустрији, у области *3D* површинске конструкције и *NC* програмирања.

Недуго затим је Иван Сатерленд (енг. *Sutherland*)<sup>1</sup> развио тзв. *SKETCHPAD* систем у *МИТ*-у 1969. године. Посебна одлика *SKETCHPAD*-а је била то што је допуштао да креатор графички утиче на компјутер: пројектант се може убацити у компјутер цртањем на *CRT* екрану (монитору) оловком за те сврхе. То је био прототип графичке употребе интерфејса, неопходне одлике модернизације *CAD*-а. Међутим, због изузетно високе цене графичког хардвера интерактивна компјутерска графика је све до осамдесетих година остала изван могућности примене, изузимајући технолошки најмоћније организације које су је могле користити. Због тога, прву комерцијалну примену *CAD* је имао у великим компанијама аутомобилских и ваздухопловно-космичких индустрија, као и у електроници. Како су компјутери постали приступачнији, подручје апликације се проширило.

Најутицајније доба развоја *CAD*-а било је откриће *MCS (Manufacturing and Consulting Services Inc.)*, који је написао *ADAM (Automated Drafting And Machining)*, али је још битније било снабдевање кодом компанија као што су: *McDonnell Douglas (Unigraphics)*, *Computervision (CADDs)*, *Calma*, *Gerber*, *Autotrol* и *Control Data*. Кроз деценије поједини ауто и авио произвођачи имали су самостално развијене *CAD* софтвере. Неки од њих су: *Ford (PDGS)*, *General Motors (CADANCE)*, *Mercedes-Benz (SYRSCO)*, *Nissan (CAD-I)*, *Toyota (TINCA)*; *Lockheed (CADAM)*, *McDonnell-Douglas (CADD)*.

Предност програмирања и рачунарског софтвера, приметно поузданог моделирања осамдесетих година прошлог века, допустиле су прилагодљивије компјутерске апликације у дизајнирању. Кључни производи за 1981. годину су били пакети поузданог (чврстог) модела – *Romulus (ShapeData)* и *Uni-Solid (Unigraphics)*, по узору на *PADL-2* и избацавање површинских моделера *CATIA (Dassault Systemes)*. Године 1982. Џон Вокер је пронашао *Autodesk*, који је водио у *2D* систем *AutoCAD*. Следећа прекретница била је стварање *Pro/Engineer* програма 1988. године, који је наговестио већу употребу одлика основних метода моделирања и који деведесетих година постаје лидер у *CAD/CAM/CAE* апликацијама. Године 1992. формиран је *CoCreate Software* као део *Hewlett-Packard* компаније са задатком да развијају *CAD* и *CAM* софтвер.

Од 1991. године *Microsoft* развија *Open GL* за *Windows NT*. *Open GL* је *API* софтвер за креирање *3D* графике и садржи команде за цртање тачака, линија, итд. Такође садржи и подршку за сенке, текстуре, анимације, светлост, ефекте – као што је магла, те за разне друге симулације.

---

<sup>1</sup> Он је касније креирао компанију графичке технологије са Дејвидом Евансом.

Године 1995. представљен је *SolidWorks* као јефтини конкурент *CAD* програмима као што су *Pro/ENGINEER*, *SDRC*, *I-DEAS*, *Unigraphics* и *CATIA*, а тренутно је један од најпопуларнијих производа на тржишту *MCAD* програма средње класе. Године 1996. *Solid Edge v3* из *Intergraph*-а је стигао на тржиште по цени од 6000 долара.

Почев од касних осамдесетих година, развојем доступних *CAD* програма, који су се могли користити на персоналним рачунарима, започео је тренд масовног смањивања одељења за скицирање у мањим компанијама.

У области пројектовања нових производа и технологија примена рачунара кроз увођење система као што су *CAD/CAM*, *CAE*, *CAPP*, интеграција кроз конкурентно инжењерство у пројектовању је императив у одржавању конкурентног положаја на тржишту.

Савремени *CAD/CAM* системи пружају широк избор различитих приступа у пројектовању производа и технологија.

Могући алтернативни приступи су: експлицитно, параметарско, варијантно и моделирање на бази типских облика (*Features*).

Не улазећи у све карактеристике наведених приступа, наводи се да је експлицитно моделирање стандардно присутно код свих *CAD/CAM* система и има основни недостатак – релативну нефлексибилност, посебно ако се ради о производима који у основи имају геометријско-технолошку сличност и могућност формирања по основу фамилија сличних делова.

Један од водећих светских *CAD/CAM/CAE* система је и *Solidworks® 2009* корпорације *Dessault systemes*.

Већина *CAD* програма пружа могућност израде геометријских модела, а многи од њих раде и на персоналним рачунарима. Најпознатији су *CAD* програми опште намене, а разрађени су и посебни додаци за специфичну примену. Најпознатији међу њима су: *AutoCAD*, *ProEngineer*, *SolidWorks*, *Solid Edge* и *CATIA*.

*AutoCAD* је стекао велику популарност и донео *AutoDESK*-у велику зараду, те послужио као основа за развој многих других уже специјализованих производа, међу које спадају *AutoCAD Map* (GIS program), *Mechanical Desktop*, *Inventor* (параметарско моделирање и модели од савијеног лима), *AutoCAD Architectural Desktop* (намењен архитектима), *AutoCAD Civil Design* (грађевинарство), *CAD Overlay* (дигитализација), *AutoCAD Land Development Desktop* (грађевинарство, катастар, урбанизам) и други.

Развојем бројних нових апликација и куповином постојећих, те њиховим укључивањем у своју понуду, компанија *Autodesk* ће у блиској будућности моћи понудити један заокружен систем који ће равноправно конкурисати *CAD* системима високе свеобухватности. Раним напуштањем развоја апликација под *UNIX* оперативним системом и потпуним опредељењем за платформу *Microsoft Windows* као примарну за своје програме, стечена је значајна предност у односу на остале произвођаче *CAD* система високе свеобухватности. Стога већина апликација компаније *Autodesk* надмашује паритетне апликације, како по

квалитетнијем изгледу екрана, тако и по имплементацији и брзини извођења у оперативном систему *Microsoft Windows*.

*AutoCAD* је софистицирани пројектантски алат широке, универзалне намене који подржава дводимензионално пројектовање. Њиме се замењује класично пројектовање на папиру или тродимензионално моделирање комплексних објеката који се у „простору модела” (енг. *model space*) могу произвољно зумирати, нагињати, окретати, приказивати у пројекцијама, погледима и пресецима из свих смерова, са перспективним ефектом или без њега, произвољно осветљавати и рендеровати тако да *3D* приказ имитира фотографију виртуелног објекта који постоји само у меморији рачунара. За разлику од алтернативних *2D* и *3D* система за моделирање, *AutoCAD* карактерише софистицирани (можда мало и прекомпликовани) систем мерила и висока прецизност, која може ићи и испод милимикрона, и аутоматски калкулисан систем димензионисања (котирања) мера, који задовољава и најстроже техничке стандарде. Уз извесне услове, котирање је асоцијативно, тј. аутоматски следе измене геометрије обрађиваног модела.

*SolidWorks* је распрострањен програмски пакет за рачунарско пројектовање и техничке анализе. Такође се користи у једноставним симулацијама и инжењерским анализама. Уз многе додатке, користан је у разним подручјима технике (машинство, електротехника), технологији обраде дрвета, итд. Познат је као снажан и једноставан софтверски пакет. *SolidWorks* развија компанија *SolidWorks Corporation*. Један је од првих *CAD* програма који је дизајниран за *Microsoft Windows*. Прва верзија је уведена 1993. године. Ова компанија је од 1997. године у власништву компаније *Dassault Systemes*.

*Catia* програмски пакет у почетку је имао назив *CATI (Conception Assistée Tridimensionnelle Interactive French for Interactive Aided Three Dimensional Design)*, а преименована у *CATIA* 1981. године. *CATIA (Computer Aided Three Dimensional Interactive Application)* је комерцијални програмски пакет развијен у француској компанији *Dassault Systemes* крајем 70-их и почетком 80-их година 20. века како би се унапредио развој француског војног авиона *Mirage*. Власник фирме *CATIA* је *IBM*, који је у почецима свог развоја нарочито био присутан у авионској индустрији. У верзији 5 садржана је 51 апликација, које се организују у две платформе – П1 и П2. Платформа П1 је намењена мање захтевним корисницима, док је платформа П2 усмерена на велика производна предузећа. Унутар сваке платформе могуће је одабрати једну од више понуђених конфигурација система, а на одабрану конфигурацију је могуће додати или докупити потребну апликацију. Апликације су подељене у подручја као код *CAD/CAM/CAE* система високе свеобухватности. *CATIA* је стандардни софтвер у ваздухопловној индустрији (*Boing, Airbus*), индустрији робе широке потрошње, машиноградњи, а посебно је заступљен у аутомобилској индустрији, у којој га користе највећи светски произвођачи аутомобила (*Mercedes, VW, Audi, Peugeot, Citroën*), као и њихови добављачи.

*Solid Edge* је *3D* параметарски *CAD* софтверски систем. Има одличну базу за моделирање, погодну за различите индустријске потребе и потпуно интегрисано управљање конструкцијама (*Insight*), што помаже *Solid Edge*-у у изради прецизних решења за конструкцију. Тиме се значајно смањују укупни трошкови развоја и производње нових производа које је могуће понудити

купцима у краћем времену. Инжењерима омогућава једноставан развој производа, од основног дела до коначног склопа. Прилагођене наредбе са смисленим током креирања убрзавају конструисање, контролисање и промену елемената унутар склопа. Управљање *CAD* алатима, које инжењери свакодневно употребљавају, удружује с управљањем конструкције *Insight*, *РДМ* систем. *Insight* може да управља конструкцијом одмах након њеног креирања, а уједно помаже координацију инжењерске групе и тиме елиминише грешке које се јављају услед лоше комуникације.

***Pro/Engineer*** или кратко *Pro/E* је тродимензионални рачунарски програм за рачунарско пројектовање. Аутор програма је компанија *Parametric Technology Corporation (PTC)*. *Pro/E* користи се у машиноградњи, електротехници, грађевинарству, архитектури итд. Компанија *Parametric Technology Corporation* основана је 1985. године. Програм *Pro/E* уведен је 1988. године, а први корисник била је компанија *John Deere*. На *Microsoft Windows* платформи овај програм се примењује од 1993. године. Најпознатији корисници *Pro/E*-програма су *Audi*, *BMW*, *Boing*, *Bosh*, *John Deere*, *Volkswagen*.

*CAD* системи се могу класификовати на основу неколико критеријума:

- **Поље примене** тиче се индустријске области у којој се *CAD* системи развијају – машинство, електротехника, архитектонско пројектовање, грађевинарство, тканина и дизајн обуће; медицинска примена.
- **Тип моделирања** може бити: *2D* или *3D* моделирање.
- **Тип објекта** моделирања може да буде: *жичано моделирање*, када су дефинисане само ивице дела, *површинско моделирање*, када је *CAD* модел празан и где су само границе површине дефинисане, *запремнинско моделирање*, које обезбеђује реалну представу модела који се састоји од једноставних основних елемената.

## 1.4. Методе истраживања

У раду је коришћен аналитички, синтетички и експериментални метод. Аналитички метод је коришћен за анализу постојећих језика и поступака за геометријско моделирање. За израду концепта и једне изведене варијанте у виду новог језика за геометријско моделирање коришћен је синтетички метод. Експериментални метод доминантан је у провери функционисања језика за геометријско моделирање (*2D*).

## 1.5. Значај истраживања

У најшире коришћеном пакету за пројектовање *AutoCAD* не постоји погодан језик за параметарско моделирање. Може да се користи програмски језик *AutoLISP*, али он је исувише компликован за просечног пројектанта. Развојем језика за *2D* параметарско моделирање на платформи пакета *AutoCAD* овај проблем је решен. Осим тога, поједина решења у језику за *2D* параметарско геометријско моделирање или превазилазе решења из пакета већих перформанси

или су на истом нивоу са њима. То су *SolidWorks*, *Catia*, *Solid Edge*, у делу који покрива *2D* решења за ове пакете.

Дакле, не само да је побољшан пакет *AutoCAD* него се решења из језика за *2D* параметарско геометријско моделирање могу користити и у пакетима већих перформанси. Ту се пре свега мисли на решења за дефиниције ентитета, где на основу осталог дела описа ентитета постоји више решења.

Увођењем нових елемената за опис ентитета повећава се број дефиниција за параметарско геометријско моделирање у односу на постојеће системе и језике за *2D* моделирање.

Већ је немачка фирма *Data M* показала интересовање за овај Језик за *2D* параметарско геомтријско моделирање, што се може видети у прилогу 1.

Аустралијска организација *SCIE* ставила је рад [56] на своју листу светских научних радова, што се може видети у прилогу број 2.



## 2. ТРЕНД РАЗВОЈА И УТИЦАЈ НА ИНДУСТРИЈУ

### 2.1 Место и улога рачунара у процесу конструисања

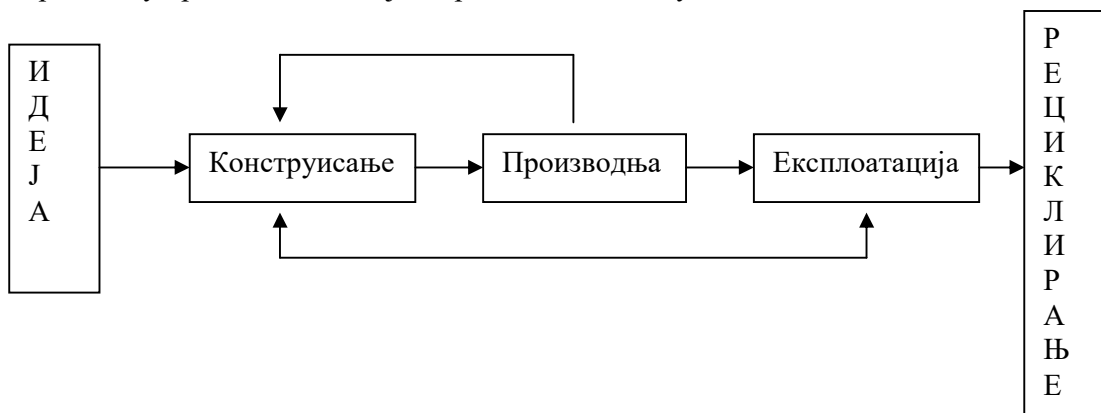
Процес пројектовања [33] представља налажење научно оправданог техничког решења које се практично може реализовати. У ужем смислу, пројектовање представља дефинисање техничког решења са функционалног, садржајног и конструктивног аспекта.

*CAD* даје следеће предности пројектовању:

- виши ниво квалитета (тачност, финоћа),
- реализује најобимније задатке и ослобађа човека рутинског рада,
- визуелна интерпретација форме која омогућава оцену функционалних и естетских вредности производа пре његове израде,
- омогућава специфичне анализе (прорачун локалних напона и сложених структура), из чега следи побољшање склопа пре израде,
- омогућава брзу израду техничке базе података, почевши од техничких цртежа до технолошких инструкција за аутоматизовану израду,
- широко интегрисана производња рачунаром (*Computer Integrated Manufacturing*).

Основни продукт пројектовања је конструкциона документација. Код класичног пројектовања она се формира на папиру и даље дистрибуира кроз техничку припрему у производњи. Код примене рачунара у пројектовању ток је исти, само је документација присутна на различитим медијумима. У принципу, код комплетних *CIM* система документација не мора бити нацртана и на папиру. Објекат пројектовања може бити производ или процес код пројектовања технологија.

Конструисање представља стварање нечег новог побољшавањем постојећих конструкција, њиховим мењањем како би обављале нове функције, или је то, једноставно речено, увођење новог концепта. Конструисање производа захтева дефинисање ограничења решаваног проблема, а затим предлагање решења које ће функционисати уз ова ограничења. Конструисање техничких система развило се из процеса израде у засебну и заокружену целину. Етапе у развоју и коришћењу производа од идеје до рециклаже дате су на слици 1.



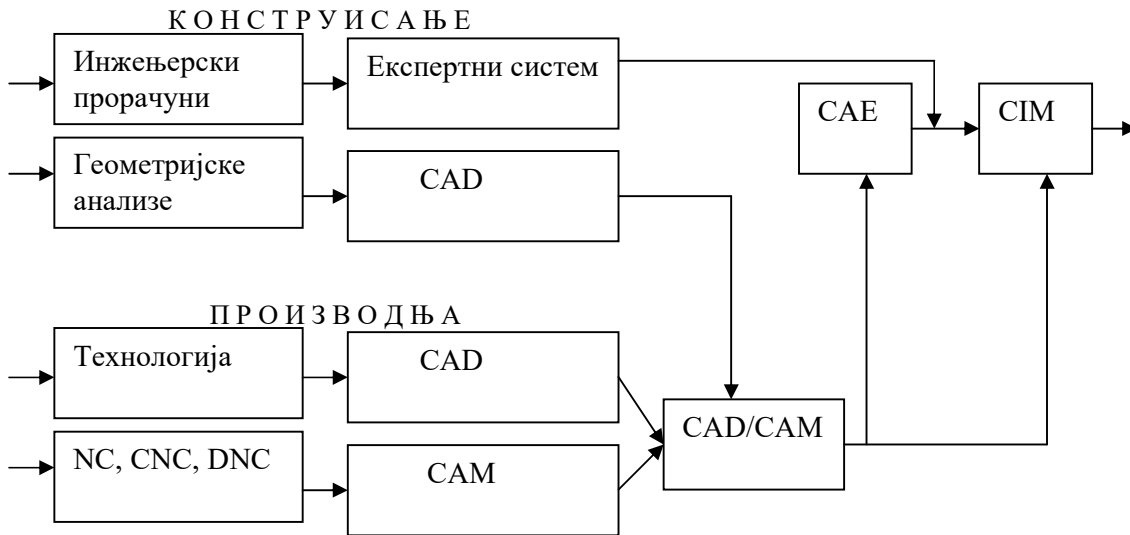
Слика 1: Етапе у развоју и коришћењу производа од идеје до рециклаже (уништења) [33]

*CAD* је стекао велику популарност последњих деценија и постао је неопходан алат за било који инжењерски задатак. *CAD* је спој човека и машине који раде заједно како би постигли оптималне конструкције у што краћем року. Графичке могућности и брзи прорачуни помоћу рачунара дозвољавају конструкторима да обликују и тестирају своје идеје интерактивно у реалном времену, а да не морају стварати праве прототипове као у конвенционалном приступу.

Како употреба *CAD* система побољшава конструисање, може се демонстрирати анализом сваког корака у конструсању. На пример, инжењер формулише проблем користећи се разним изворима информација. Са *CAD* радног места конструктор може приступити великом броју одабраних информација које се односе на тренутно одређени конструкцијски проблем.

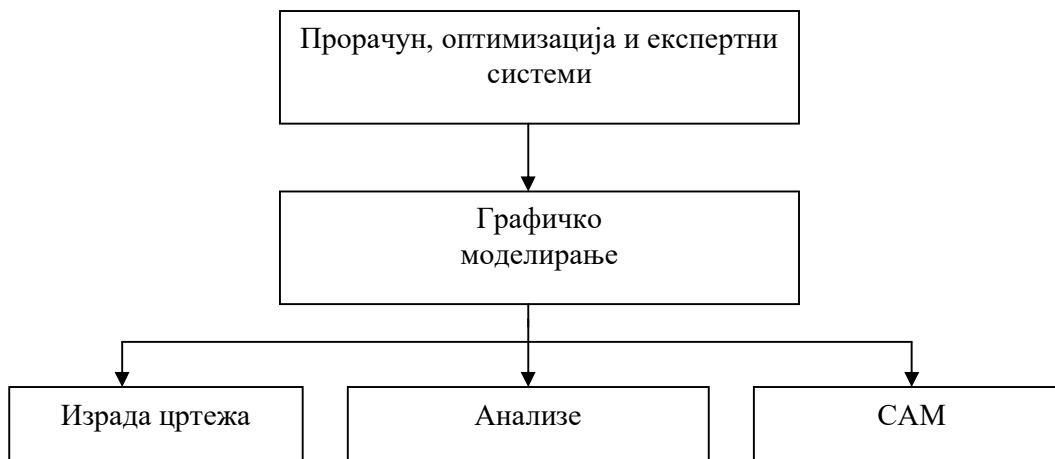
Процес конструсања у својој основи је информациони процес, јер полази од идеје, одвија се коришћењем знања и података и завршава се информацијом о томе како треба објекат изгледати и какво се понашање у експлоатацији очекује. Развитком методе обављања операција помоћу рачунара створени су и системи за конструисање, од којих су најважнији *CAD*, *CAM*, експертни системи и слично.

Наведени системи при пројектовању и конструисању производа и технологије дати су на слици 2.



Слика 2: Примена рачунара у процесу конструисања и производњи [33]

На слици 3 дат је упрошћени приказ структуре која илуструје претходни и наредни корак у процесу графичког моделирања. Димензије делова, толеранције и други параметри облика делова добијају се као резултат прорачуна, оптимизирања и слично.



Слика 3: Структура CAD-а и веза са CAM-ом [33]

Процес конструисања постаје у вишем степену аутоматизован и добија нека обележја експертног система. Формирани графички модели (облици) користе се за израду цртежа, затим за разне врсте прорачуна и анализе стања, као и за израду програма за израду делова таквог облика (CAM). За примену методе коначних елемената потребан је графички модел машинског дела, а по овој основи и метода коначних елемената може бити укључена у CAD.

Обликовање машинских делова помоћу рачунара је много ширег значаја од самог графичког приказивања. Могу се издвојити следећих неколико битних одлика оваквог начина обликовања. У првом реду, то је могућност моделирања облика, што значи компоновање облика од простијих елемената, са димензионисањем, које се заснива на одговарајућим математичким моделима. Важна компонента обликовања помоћу рачунара је „памћење” облика, тј. могућност формирања библиотеке облика који се понављају. То омогућава да се конструисање сведе на одабирање и компоновање машинских склопова ширих структура од већ разрађених и сачуваних облика делова.

### 2.1.1. Методе геометријског моделирања [33]

Овде ће се само поменути параметарско моделирање, што је предмет овог докторског рада. Већина машинских делова је варијантног и поновљивог облика. Ове моделе није рационално моделирати интерактивно, него је много делотворније користити параметарско моделирање, где се моделира на један од два начина и чува у виду програма.

Један од језика са којима се може вршити параметарско моделирање је *AutoLISP* у оквиру пакета *AutoCAD*. Уз помоћ *AutoLISP*-а могу се аутоматизовати радње које се често понављају и могу се изградити програми за параметарско цртање. Једноставан пример параметарског цртања је команда *Insert*. Након позивања команде *Insert* и уноса имена блока, могу се задати фактори промене

његове изворне величине за сва три смера координатних оса, као и угао ротације. Претходно се командом *Block* у пакету *AutoCAD* памти блок који се позива командом *Insert*. Међутим, ако желимо да мењамо димензије које нису дате у блоковима који су сачувани, не можемо користити овај начин параметарског моделирања. Параметарско моделирање заснива се на погодности да се између димензија једног машинског дела успоставе међузависности, односно да се изразе преко неколико величина које представљају параметре. Тиме је омогућено да се облик дефинише рачунарским програмом за моделирање машинског дела у којем су улазне величине називне мере. Највећи број машинских делова се може дефинисати преко неколико величина, тј. димензија. Неке од њих су резултат прорачуна, а неке дефинишу однос овог дела у односу на друге у склопу.

### 2.1.2. Процес конструисања (структура и окружење) [33]

Процес конструисања је сложен технички, теоријски, експериментални и информациони процес, који се одвија у сложенем друштвено-технолошком окружењу. Нове методе у процесу конструисања, које су се нагло развиле са развојем рачунарске технике и *CAD* система, нису раздвојене од традиционалних. Сматра се да су нове методе резултат само једне фазе у развоју процеса конструисања, подстакнуте развојем нових средстава да би се неки делови процеса ефикасније и брже извели. Структура *CAD* софтвера може бити једнокомпонентна или вишекомпонентна, зависно од сложености третиране проблематике. Осим тога, на то утичу и потребе које *CAD* систем треба да обезбеди. *CAD* софтвер треба да омогући синхронизовано и аутоматизовано одређивање процеса пројектовања и осталих активности, при чему степен аутоматизације може бити потпун или остварен до одређеног нивоа. Вишекомпонентни софтвер састоји се из тзв. *модула*, који могу бити основни и посебни.

У основне модуле спадају:

- конструкцијски модул, који обезбеђује комуникацију између пројектанта-конструктора и рачунара, односно *CAD* система,
- геометријски моделар за графичку презентацију у простору *3D* или раванском *2D* облику,
- јединствена база података са геометријским моделима компонената, саставницама, стандардним деловима и слично,
- модул за комуникацију са другим *CAD* системима и
- модул за израду техничке документације.

Посебни модули који могу бити у *CAD* систему су:

- модул за креирање концепта производа,
- модул за анализу и симулацију механизма,
- модул за управљање датотекама,
- модул за површинско и запреминско моделирање,
- модул за нумеричи управљане машине, тј. *NC* апликације.

Софтвер за графичку презентацију постао је разноврстан и поред настојања да се стандардизацијом ограниче различити приступи и некомпатибилност.

У тој разноврсности могу се издвојити софтвери за:

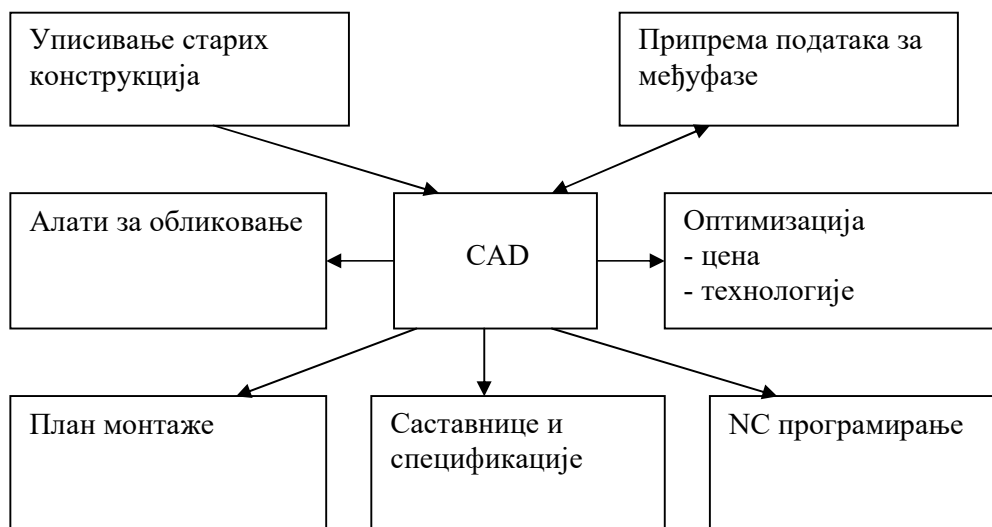
- графичку презентацију скупова тачака које представљају резултате прорачуна и једноставнијих линија и контура,
- 2D презентацију (може се рећи да је ово алат за израду цртежа),
- просторну 3D презентацију, који омогућавају формирање графичких приказа који пружају релативно велике могућности, тј. омогућавају увођење принципа за аутоматизацију процеса конструисања, а у даљем кораку стварање интелигентних система за одвијање овог процеса.

### 2.1.3. Процесори, препроцесори и постпроцесори [33]

Сам *CAD* је највећим делом електронски обликована табла за цртање. Међутим, на ово се може додати још много компонената. То су, пре свега, препроцесори и постпроцесори, који проширују *CAD* у многим правцима. Слика 4 приказује захтеве према *CAD* системима и њиховом околином.

Процесори су самостални делови програма који дају специфичне излазе. Препроцесори су програми који предају податке *CAD* системима, а постпроцесори су програми који прерађују податке из *CAD*-а.

Према својој функцији у аутоматизацији производње у области *CAD*-а, постоји знатно више постпроцесора него препроцесора.



Слика 4: Околина *CAD* и потреба за препроцесорима и постпроцесорима [33]

#### 2.1.4. Нумерички управљане машине алатке – NC програмирање [33]

Конструисање помоћу CAD-а обезбеђује минимум геометрију производа. NC машине, као што су стругови, глодалице, пресе, ласерске машине за сечење и слично, користе управо то за дефинисање обратка. Процесор решава технолошки процес (редослед резања, алате, путање алата и слично). Овај програм прелази у неки други формат применом NC машине, помоћу једног новог постпроцесора који одговара том типу NC машине. Код повезивања NC машине са CAD системима разликују се два модела. Први модел се односи на ситуацију кад је примени рачунара у пројектовању производа претходила примена рачунара у пројектовању технологије за NC машину алатку. Други модел повезивања CAD система са NC програмирањем омогућава опис обратка у облику NC управљачких информација са графичком интерактивном техником, када CAD систем омогућује добијање NC геометрије на основу формираног модела обратка у рачунару.

### 2.2. Уопште о CAD системима и параметарском моделирању

У CAD-у се све више ради на интеграцији активности које су раније биле развојене. У првим годинама било је доста полемике око природе и карактеристика CAD-а. На пример, да ли „А” стоји за аутоматизацију или за помоћ (*A-automated,-aided*). У почетку се покушавало да се процес пројектовања аутоматизује стварањем програма који су требали да копирају нумеричке и логичке покушаје које су чинили пројектанти. Овакав приступ је имао много недостатака, а најважнији је неодговарајуће коришћење човековог начина расуђивања код пројектовања помоћу рачунара.

Суштина је да се контролише целокупан след решавања проблема. Овај интерактивни концепт партнерства између пројектанта и рачунара допуњен је проширењем меморије, повећањем брзине рада и визуелног приказивања, тако да пројектант види оно што ради. Постоје и системи који имају могућност да им се постављају питања у разним базама података. Паралелни развој база података и софтверске графике за визуелно приказивање омогућило је развој повезаних радних станица. Мрежно повезивање радних станица омогућило је приступ заједничким информацијама.

Ограничења у хардверу су све мања, а једино питање је остао софтвер. Приступ софтверском систему је био да се развије модел онога шта је потребно, а затим да се такав модел разбије на модуле и интерфејсе. Извршење оваког задатка је захтевало много напора и времена, а њега нису могли да реше само програмери, већ је била неопходна и сарадња са инжењерима.

У машинству, и уопште у производној области, напредак је био спорији. Утврђено је да се већи део трошкова у свакој компанији јавља у производном делу посла, а да се мање од 5% јавља у току пројектовања, тако да CAD не би много допринео укупном смањењу трошкова и било каква велика улагања у овој области не би се исплатила. Овај, у основи погрешан закључак, који је имао негативан утицај на почетну подршку и развој, сада је обрнут, јер је чињеница да се 85% од укупних трошкова у стварности утврђује у фази пројектовања, а то је од кључне важности за остварење профита.

Да би се суочили са растућом конкуренцијом на интернационалном плану и резултујућом глобализацијом тржишта, произвођачи се морају укључити у већ

започету потрагу за новим техникама, методама и алатима. Данас се у пракси често дешава да је потребно извршити модификације на пројекту, што има за последицу скупа кашњења код предузећа која не користе предности савремене технологије везане за *CAD*. Да би се постојеће и нове технологије у компанијама ефикасно користиле, тада организација треба да успостави боље везе на релацији маркетинг – пројектовање – производња – продаја. Кључни захтеви су да се, са једне стране, остваре што боља својства и карактеристике производа, као и нижа цена, а са друге стране, да се постигне краће време пројектовања. Према томе, све више се наглашава потреба за снажнијим *CAD* системима. При том постоји потреба да се у пројектним активностима уносе до извесног степена и нове идеје у производ.

Постојећи *CAD* системи нису у могућности да задовоље један део захтева које постављају стварни пројекти јер је у њима садржана углавном конвенционална технологија обраде информација.

Задаци који се постављају код пројектовања *CAD* система требало би да укључују следеће:

- човек пројектант мора да врши контролу процеса пројектовања, али, у исто време, *CAD* систем мора да буде у могућности да узме учешћа у том процесу што више,
- пројектовање је активност кроз коју се долази до производа који задовољава постављене захтеве. Међутим, у стварним условима пројектовања таквог производа ови захтеви нису увек на самом почетку фиксни, већ се мењају чак и при изради пројекта,
- пројектне алтернативе не смеју да буду унапред много ограничене,
- предмет који се пројектује може да буде веома сложен и стога *CAD* систем мора да буде у могућности да обрађује много података и да се предмет може прецизно презентовати, а у исто време да се може флексибилно манипулисати подацима,
- пројектовање је сложена активност у којој се јављају различити типови проблема и различити типови података, тако да *CAD* систем мора да буде такав да може да пружи подршку свим аспектима пројектовања, као и да се у њему може, код пројектовања, радити са свим типовима података,
- код пројектовања у коме се користи много стандардних и типских елемената и форми неопходан је систем за параметарско моделирање (оваквих елемената и форми).

Системи параметарског моделирања користе се пре свега код пројектовања фамилије елемената, код којих се поједини чланови фамилије разликују само по димензијама. Коришћењем система параметарског моделирања, циклус пројектант – производња може лакше и брже да се спроведе.

Параметарски језици и системи за моделирање морају имати следеће карактеристике:

- 1) једноставни за коришћење;
- 2) недвосмислени;
- 3) брзи;
- 4) прилагођени специфичним захтевима и
- 5) нису превише сложени.

Све то значи да *CAD* системи морају задовољити услове као што су: могућност прилагођавања, могућност проширења, што већа брзина пројектовања, практичност и општост. Оптимално *CAD* решење мора да омогућава брзо креирање и уређивање геометрије. Немогућност да се изврши брзи преокрет, може озбиљно утицати на производњу и успех производа.

### **2.2.1. 2D CAD системи**

Једноставни *2D CAD* системи су векторски цртачки програми. Елементи цртежа су тачке, линије, кружнице и лукови, те криве линије. Различити алати омогућују стварање, позиционирање, измену и брисање ових елемената. Начин рада се незнатно разликује од начина рада на класичној дасци за цртање. Осетне разлике се постижу коришћењем слојева и унапред дефинисаним симболима (нпр. за стандардне и њима сличне делове и елементе који се понављају). Компликовани прорачуни тачних мера више нису потребни, јер су *CAD* програми знатно прецизнији од класичних ручно израђених цртежа. Унапређени *CAD* системи подржавају полуаутоматско и потпуно аутоматско цртање кота и шрафура. Следећа карактеристика савремених *2D CAD* система је коришћење асоцијативности између појединих елемената цртежа, на пример између линије и коте. Снажни *CAD* системи стављају на располагање и могућност додавања властитих функција за лично прилагођавање (разним програмским језицима).

### **2.2.2. Параметарски модел**

Овом се методом стварају асоцијативни параметри из запремина, површина, тела и услова. Сам модел постаје променљив према параметрима. На пример, један завртањ који је једном конструисан може се уз мењање параметра пречника навоја одмах претворити у произвољно велики завртањ. Овако се штеди много времена, простора за складиштење и омогућује стварање ефикасних библиотека стандардних и њима сличних делова.

#### **2.2.2.1 Параметарско и варијацијско пројектовање [19]**

У многим случајевима дизајнер хоће да креира производ са геометријом која представља варијацију неких претходних конструкција. Многе компаније имају производе који су делови стандардизованих варијација одређених фамилија производа или који користе геометрију претходно дизајнираних производа, али са променама у димензијама. Често, у раним фазама дизајнирања нових производа, дизајнер зна шта је потребно, али не може тачно одредити димензије, или пак жели да истражује варијације у димензијама како дизајн напредује. У сваком случају, објекти су потребни да би се креирале варијације облика у нешто што је познато као *параметарски дизајн*.

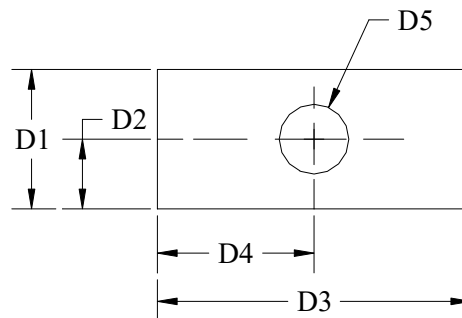
#### **2.2.2.2 Класификација приступа параметарском дизајнирању**

Постоје препоруке да се параметарски дизајн сврста у варијанте програмирања, експертских система, конструктивних шема и да се поспешује њихово решавање. Решавање нумеричких ограничења дозвољава да се дефинишу



димензије дела, у смислу односа и ограничења, и да се генеришу варијанте делова применом неке врсте алгоритма који задовољава задата ограничења. Разлика између параметарског и варијацијског приступа је у типу примењеног алгоритма који задовољава задата ограничења. Параметарски системи решавају ограничења варијабилних модела по реду (секвенцијално), где се свака додељена вредност рачуна као функција раније додељене вредности. Варијациони системи изграђују систем једначина, које представљају ограничења, а затим решавају све једначине истовремено користећи процедуре нумеричког решавања или неке еквивалентне методе. И конструктиван и ограничавајући приступ решавања називају се параметарским моделирањем у комерцијалним системима. Параметарски и варијациони модели користе скуп једначина за одређивање величине и облика неког машинског дела.

Разлика између параметарског и варијацијског приступа код моделирања је дата у [69].



Слика 5: Пример параметарског објекта [69]

Ограничења код моделирања објекта датог на слици 5 су:

- дужина објекта треба да буде два пута већа од његове ширине,
- дубина објекта треба да буде половина ширине објекта,
- отвор треба да се налази у самом центру,
- пречник отвора треба да буде 0.6 од његове ширине.

Сада се може погледати шта се подразумева под параметарским и варијационим моделирањем. Пројектант (конструктор) може да наметне одређена ограничења, обично не дајући све димензије геометрије. Током процеса моделирања, као у овом примеру, дужина објекта треба да буде два пута већа од његове ширине. Ово ограничење, које намеће пројектант, у облику је једначине. Постоји и друга намера да дубина буде једнака половини ширине објекта. Ово је друга намера која долази због одређених функционалних аспеката објекта. Онда отвор треба да се налази у самом центру и даје се ограничење да пречник отвора мора да буде 0.6 од ширине. Овим се пројектант ограђује од тога да отвор не буде такав да зид између отвора и бочних површина не буде сувише мали.

Сада је потребно да се пројектант побрине да ових шест параметара на слици 5 (са дужином  $D_6$ ) дефинишу објекат заједно са намером пројектанта. Број

параметара који је довољно ефикасан може бити мањи. Не мора се обезбедити свих 6 вредности како би се дефинисао објекат, јер ограничења ће практично смањити број улазних параметара.

Често ограничења може бити превише, па услед тога ти конфликти могу да доведу до две ситуације. Дакле, овде постоје два ограничења која су конфликтна, и оба се не могу задовољити.

Код пројектовања објеката може постојати *вишак* ограничења или *мањак* ограничења. Ако је мањак ограничења, онда нема довољно ограничења које софтвер мора да препозна да би се објекат дефинисао јединствено.

*Параметарски моделари* користе итеративне стратегије са ограниченим могућностима ограничења. Параметарски системи се могу користити за моделирање широког спектра уобичајено конструисаних делова. Они су једноставни за коришћење и не дозвољавају ненамерне геометријске инверзије или промене у топологији. Секвенцијално решење моделирања дозвољава да се аутоматски евидентира дизајн историја.

Параметарско моделирање има ограничења која су секвенцијално дефинисана. Свако ново ограничење се користи после претходно дефинисаних ограничења, тако да је важан њихов редослед.

Дакле, у параметарском моделирању намера пројектанта мора бити написана по тачно утврђеном редоследу. Идеја је да се избаци једно ограничење, а то зависи од приоритета. Оно ограничење које има највиши приоритет мора се поштовати, док ограничење са нижим приоритетом мора бити нешто што се може по потреби променити.

Свако ограничење се израчунава на основу претходно дефинисаних ограничења.

Ограничења пројектанта приказана на слици 5 су:

- $D_1 = x - x$  је улаз који мора да се обезбеди, а то је ширина објекта,
- $D_2 = D_1/2$  – ако се дефинише да је  $D_2$  једна од димензија које дефинишу центар отвора тако да буде на растојању половина ширине  $D_2 = D_1/2$ ,
- $D_3 = 2 * D_1$  – параметар  $D_3$ , који је дужина објекта, два пута је већи од ширине објекта  $D_3 = 2 * D_1$ ,
- $D_4 = D_3/2$  – овде је друго растојање од ивице објекта  $D_4$  и оно је једнако  $D_3/2$ ,
- $D_5 = 0.6 * D_1$  – дефинисано је да је дубина објекта једнака половини ширине објекта. И још једно ограничење је да пречник отвора  $D_5$  треба да буде 0.6 пута од вредности  $D_1$  која је дефинисана,
- $D_6 = 0.5 * D_1$  – дубина.

Када се да један улаз  $x$ , могу се израчунати све остале вредности. Дакле, догодило се то да се шест ограничења сведу на једно. Тиме је облик и величина решена.

*Варијациони моделари* нису ограничени на специфичне класе производа (као што су параметарски моделари) и решавају сет једначина истовремено. Једначине за дизајн треба да буду такве да задовољавају сва ограничења. Време израде модела са конкретним димензијама је дуже од параметарских моделара и решења могу резултирати у нежељеном правцу конфигурације геометрије. Постоје и хибридни системи. Разлика између софтверских пакета није јасна. Дизајнер треба да одабере систем који је способан да моделира одређене делове.

*Варијационо моделирање* – ограничења решава истовремено и редослед спецификације ограничења није важан. Сва ограничења се дефинишу као једначине које се решавају истовремено.

Ограничења пројектанта су:

$$D_1 - x = 0$$

$$D_1 - 2 * D_2 = 0$$

$$2 * D_1 - 2 * D_4 = 0$$

$$0.6 * D_1 - D_5 = 0$$

$$0.5 * D_1 - D_6 = 0$$

У моделирању се користе тригонометријски односи, квадратни корен, полином, итд. Решење понекад није могуће, а то значи да није могуће решити ове једначине аналитички.

Код параметарског моделирања решење се изводи лако, идући секвенцијално кроз ограничења, што је много једноставније. Са друге стране, код варијационог моделирања не морају се дати ограничења у одређеном редоследу, али се до решења једначина долази преко нумеричких метода, итд.

Ограничења која су до сада поменута могу да буду проширена са геометријским ограничењима. Две линије могу да буду паралелне, управне, под одређеним углом, и сл. Тако да имамо две врсте односа. Једна су димензионална ограничења, а друга су геометријска ограничења. Пројектанту је боље дати геометријска ограничења, јер је то удобније за рад, него да димензије решава сам. Дакле, дизајнер не мора решити коначно сву геометрију производа, решење може да да и софтвер. Пројектант даје само оне намере које треба да буду задовољене као део дизајна. Ово је моћан аспект последњих *CAD/CAM* система, где се користи параметарско или варијационо моделирање.

### 2.2.2.3 Систем описивања конструкцијских елемената

Овде постоје три различита приступа описивања делова помоћу карактеристика који морају да функционишу у тесној вези са *CAD* системом.

Код **првог** приступа (дизајн помоћу карактеристика) корисник може да опише параметарске делове користећи каталог карактеристика у *CAD* систему (када *CAD* систем није параметарски). Он може интерактивно да изабере карактеристику, убаци њене особине и лоцира је у делу.

Код **другог** приступа (уношење карактеристичних података) корисник уноси вредност параметара директно у систем за описивање делова (за које се описивање путем карактеристика може лако урадити). Податак (вредност параметра) се онда преноси у *CAD* систем, а конструкција дела се појављује аутоматски.

Код **трећег** приступа (параметарски дизајн) се користи параметарски *CAD*, сви параметри дела из *CAD* система се могу пребацити у описни систем дела. Сви параметри које користи аутоматско закључивање морају да се претходно прегледају у параметарском *CAD* моделу.

Када параметарски модел за део породице већ постоји, све што корисник треба да уради је да изабере модел породице дела и промени вредност параметара.

Системи параметарског пројектовања користе се пре свега код пројектовања фамилије елемената, код којих се поједини чланови фамилије разликују само по димензијама. На овај начин циклус пројектант–производња може лако и брзо да се спроведе. Поготову што у почетној фази пројектовања димензије обично нису познате.

Постоје развијене технике како се то решава у *CAD*-у. Притом, могу се уочити два основна приступа:

- програмирање са димензионалним параметрима на неком од програмских језика високог нивоа,
- графичко интерактивно пројектовање главног модела и касније аутоматско добијање димензионалних варијанти на основу главног модела.

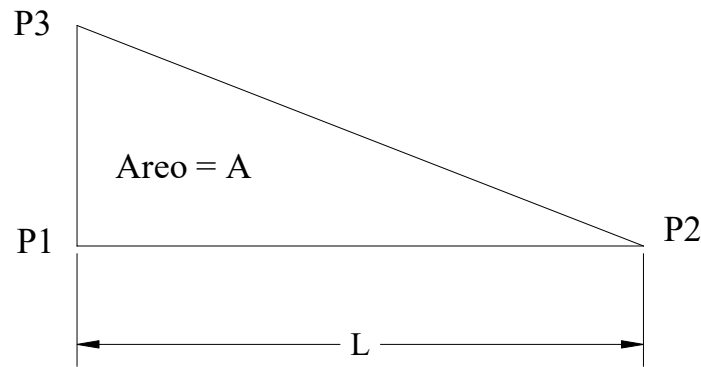
Први приступ захтева да корисник поседује специфично програмско знање. Други приступ (метода графичког интерактивног пројектовања) заснива се на оперисању са имплицитним ограничењима, као што су тангенцијални однос, управност, паралелност итд. Међутим, ово још није решено на задовољавајући начин. Ручно одређивање групе имплицитних граничних услова (ограничења) је споро и доводи до грешака. Ако се примене методе аутоматског препознавања имплицитних ограничења, потребне су дуготрајне ручне провере да би се пречистила случајно развијена ограничења.

Генерално, важан циљ је да се *CAD* системи начине интелигентнијим у подршци при одлучивању код пројектовања.

#### **2.2.2.4 Ограничења код моделирања [47]**

Систем ограничења у бази омогућава познавање положаја и величине свих ентитета, а обично се користе променљиве. Ово омогућава дизајнеру да створи геометријски модел који се може лако мењати у току процеса пројектовања. У случају једноставних *2D* параметарских модела, за пример је узет троугао одређен са три темена  $P_1$ ,  $P_2$  и  $P_3$  (слика 6). Шест једначина је потребно да у потпуности одреде  $x$  и  $y$  координате трију тачака (у *2D*).

У бази података постоје информације у вези са повезивањем, као и врсте ентитета у вези са сваком тачком или чвором у моделу, а затим се то користи за генерисање жељене геометрије (сегмената линија за троугао), али је основна геометрија у основи дефинисана положајем скупа тачака или темена. У свом најједноставнијем облику, једначине за троугао би се састојале од доделе вредности за сваку координату (тачка  $P_1$ :  $x_1=5$ ,  $y_1=4$ , итд.). У овом случају не постоје ограничења или однос између ентитета и модел је фиксни без варијационе или параметарске способности. Коришћење променљиве уместо константи у додели једначина омогућава моделу мале варијационе способности. Ово доводи до врло флексибилног и врло општег модела троугла, али је веома незгодно модел користити за потребе пројектовања. Ограничења су потребна за контролу облика и укључују претпоставке које су обично у дизајну.



Слика 6: Модел троугла за пример ограничења

Врсте ограничења су:

*Ограничења на моделу* могу да се класификују у четири врсте: основна ограничења, димензионална ограничења, геометријска ограничења и алгебарска ограничења.

1. Локација одређене тачке и хоризонталних и вертикалних ограничења на линијским сегментима су типични примери *основних ограничења*.

2. *Димензионална ограничења* дају нумеричке вредности за основне геометријске ентитете. На пример, дужина једне стране троугла може се одредити коришћењем једначине:

$$(x_2 - x_1)^2 + (y_2 - y_1)^2 = L^2$$

где је  $L$  димензија или дизајн параметар за дужину линије сегмента  $P_1P_2$ . Параметарски системи за моделирање омогућавају корисницима да варирају ове димензије дизајна. Димензионална ограничења се могу користити да означе обим специфичног ентитета, као што су сегмент линије или полупречник лука, или могу бити примењене на било које две тачке на моделу да контролишу растојање између ових тачака.

3. *Геометријска ограничења* дају односе између геометријских ентитета, као што су: додирне тачке, паралелност, симетрија итд. Ако су две линије дефинисане преко тачака  $(x_1, y_1)$ ,  $(x_2, y_2)$  и  $(x_3, y_3)$ ,  $(x_4, y_4)$  управне, то је довољно да се да једначина облика:

$$(y_2 - y_1)/(x_2 - x_1) = -(x_4 - x_3)/(y_4 - y_3)$$

и то је потребно да се жељена ограничења наметну.

4. *Алгебарска ограничења* намећу ограничења у димензијама дизајна у облику математичких једначина. Дизајнер ће можда желети да једна линија са дужином једног сегмента буде једнака два пута дужини другог сегмента линије. Алгебарска ограничења могу бити једнакост или неједнакост. Неједнакости се могу користити да ограниче величину ентитета, или у апсолутном износу или у поређењу са другим геометријским ентитетом. Логички изрази (*IF-THEN*) дозвољавају дизајнеру да развије сложеније односе између геометријских параметара.

Све четири врсте ограничења (основно, димензионално, геометријско и алгебарско) могу се користити да се створи дизајн намера. Нека ограничења могу да буду аутоматски наметнута од стране софтвера на основу претпоставки, као што су хоризонтално или вертикално постављене линије, итд. Уопште, *CAD* системи треба да омогућавају да дизајнер одреди димензионалне вредности, геометријска ограничења између ентитета и записивање алгебарских израза. На основу овог уноса дизајнера, систем генерише потребне једначине ограничења за поједине тачке.

На пример, ако размотримо троугао са теменима  $P_1$ ,  $P_2$  и  $P_3$  приказан на слици 6, страница троугла  $P_1P_2$  има дужину  $L$  и троугао површину  $A$ . Потпуно ограничење троугла захтева шест једначина. Са претпоставком да су две странице троугла хоризонталне и вертикалне може се дати шест једначина:

$$\begin{aligned}x_1 &= 5, & y_1 &= 4, \\y_2 &= y_1, & x_3 &= x_1, \\(x_2 - x_1)^2 + (y_2 - y_1)^2 &= L^2, \\(x_1y_2 + x_2y_3 + x_3y_1 - y_1x_2 - y_2x_3 - y_3x_1) &= 2A.\end{aligned}$$

Прве четири једначине представљају основна ограничења, пета је димензионално ограничење и последња једначина је алгебарско ограничење.

Овде се даје један пример параметарског описа једне једноставне форме из литературе [47].

Задатак генератора конструкционог плана је да током процеса обликовања конвертује корисникове улазе у један формат који се назива конструкциони план, а који садржи како процедуру обликовања, тако и податке за дату форму. То се постиже помоћу генерисања променљивих величина свих координата оних тачака које карактеришу геометријске ентитете, тј. крајње тачке дужи и лукова и централне тачке кружница. То се изводи у тренутку када корисник кроз команду обликовања уноси неки нови геометријски ентитет. Затим, ту се врши конверзија свих команди за обликовање, било у фиксном, променљивом или флексибилном начину рада, у један редослед израза који обухвата дефиниције променљивих вредности координата, као и геометријске команде, при чему се као параметри појављују само променљиве вредности координата и/или фактори понављања.

На слици 7 приказана је једна проста форма за пример команди и функције генератора конструкционог плана. Форма на овој слици се састоји од две граничне линије  $L_1$  и  $L_2$  и две конструкционе линије  $C\_L_1$  и  $C\_L_2$ ; тачке су овде симболички означене са  $P_1$ ,  $P_2$  и  $P_3$ .

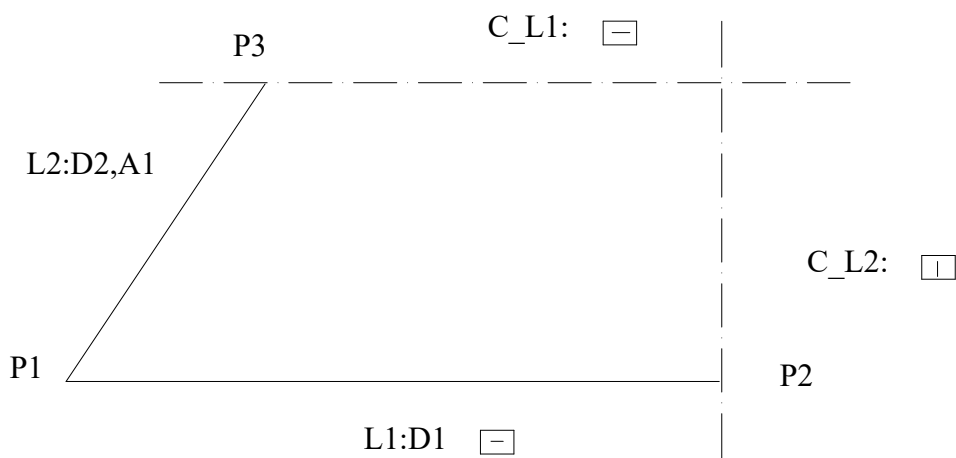
Команде обликовања и параметри команди који током корисничког рада представљају улаз су следеће:

LINE\_HORIZONTAL (почетна тачка P1, дужина D1)  
 LINE\_POINT\_LENGTH\_ANGLE (почетна тачка P1, дужина D2, угао до  
 хоризонтале A1)  
 C\_LINE\_HORIZONTAL (ниво P3)  
 C\_LINE\_VERTICAL (ниво P2)

Генератор конструкционог плана конвертује ове улазне податке у следећи облик и редослед:

```

X1=start_point_x_coordinate
Y1=start_point_y_coordinate
X2=X1+D1
Y2=Y1
LINE_BETWEEN_TWO_POINTS ((X1, Y1),(X2, Y2))
X3=X1+(D2*cos(A1))
Y3=Y1+(D2*sin(A1))
LINE_BETWEEN_TWO_POINTS ((X1, Y1),(X3, Y3))
C_LINE_POINT_ANGLE ((X3, Y3), 0)
C_LINE_POINT_ANGLE ((X2, Y2), 90)
  
```



Слика 7: Пример конструкције помоћу граничне геометрије, укључујући конструкционе линије

Из основног облика насталог конструкционим планом могу се генерисати варијанте, и то у складу са вредностима одређених димензија и почетним тачкама. Ово изводи процесор варијанти, у чији рад је још укључено давање налога за измене променљивих и конструкционог плана, у складу са прихватљивим улазом у одређени CAD систем. Притом се конструкциони план може претпроцесирати, а резултати меморисати за дизајн команде, односно параметре које траже команде у конкретном CAD систему.

## 3. ЈЕЗИК ЗА 2D ПАРАМЕТАРСКО ГЕОМЕТРИЈСКО МОДЕЛИРАЊЕ

### 3.1. Општа структура

Сваки основни графички ентитет описује се скупом који се састоји од подскупова, односно елемената за опис ентитета. Скуп (колекција) елемената за опис ентитета може се дати у општем случају као

$$S = \{S1, S2, S3, S4, S5, S6, S7\}.$$

Опис неког основног ентитета може се груписати и на следећи начин: идентификација, ознака, примарни опис ентитета (геометријски опис), секундарни опис ентитета (додатни опис).

За опис једног основног ентитета, у прве две од ове четири групе, узима се:

- S1- из скупа симбола за идентификацију ентитета,
- S2- из скупа симбола за врсту, тј. ознаку ентитета.

Опис S3 из групе за геометријски опис постоји само код ентитета који се дефинишу функцијом. S3 је једна или више функција којом се одређује кодомен функције на задатом домену. На пример, у равни (2D) – линијски (жичани) опис, S3 је једна или две функције, облика  $y=f(x)$  или  $x=f(y)$ , односно параметарског облика  $y=f1(t)$ ,  $x=f2(t)$ .

Помоћу ових функција сваком елементу из области дефинисаности се придружује једна вредност из скупа вредности.

Даље, у групу за дефинисање примарног описа када се основни ентитет не описује функцијом улази S4 као скуп подскупова координата тачака, самих координата тачака и других карактеристичних геометријских параметара, као и релационих исказа.

Колекција у скупу за примарни опис зависи од врсте ентитета. Ако основни ентитет није из групе стандардних (уобичајених) ентитета, у S4 се даје домен у којем се израчунава функција задата изразом у S3.

Код дефинисања вредности било које тачке (скупа њених координата), координата тачака, као и других карактеристичних параметара, могу да се користе изрази у којима постоје стандардне математичке операције (сабирање, одузимање, множење, дељење, степеновање), стандардне математичке функције (синус, косинус, тангенс, логаритам итд.), варијабле, константе, координате карактеристичних тачака, скупови координата тачака и други карактеристични параметри (пречник, полупречник, угао, дужина, итд.) ентитета, као и дистанца између дефинисаних тачака и координате тачака пресека ентитета. Осим тога, може се код описа вредности неке карактеристичне тачке њој придружити вредност подскупа координата неке од карактеристичних тачака било ког другог ентитета.



И код дефинисања ентитета функцијом, израз за функцију (домен и кодомен) може се формирати на исти начин као код дефинисања вредности карактеристичних координата и других карактеристичних геометријских параметара. Релацијама (тангентно, управно, паралелно, итд.) описује се положај ентитета који се описује према другом ентитету.

Специфицирањем релационих исказа тип “1” (“нешто” веће од “нечега” или “нешто” мање од “нечега”) даје се, осим тога, однос положаја тачака ентитета који се моделира (коришћењем координата тачака и других карактеристичних геометријских параметара у релационом исказу) [60,63]. То су релациони искази, где са леве и десне стране релације (веће, мање или релације које садрже релације веће или мање) постоје изрази у којима се користе координате и други карактеристични параметри ентитета. Израз са леве као и десне стране релација се специфицира на исти начин као код специфицирања координата карактеристичних тачака, или карактеристичних геометријских параметара ентитета.

Код примарног (геометријског) моделирања може се специфицирати и друга врста релационог исказа тип “2”, а то је тип исказа где само са једне стране релације (веће, мање, није веће, није мање) постоји израз, а са супротне не постоји ништа [60,63]. Овај израз, са једне стране релације, специфицира се на исти начин као и израз код релационог исказа тип “1”. Ово је релациони исказ типа “нешто” веће, ”нешто” мање, ”нешто” није веће, ”нешто” није мање и слично томе, односно, такав исказ дефинише једно од два могућа стања за “нешто”.

Ове две врсте релационих исказа користе се, код моделирања ентитета, када на основу осталог дела поступка примарног (геометријског) моделирања ентитета постоји више решења, тј. више скупова тачака за дати ентитет. У том случају овим се исказом специфицира који од тих скупова тачака треба да се издвоји. То се решава укључивањем координата и других карактеристичних параметара у релациони исказ. Код релационог исказа тип “2” најлакше се то решава захтевом да вредност неке координате, неке карактеристичне тачке или вредност неког параметра буде већа или мања (нпр.  $x$  координата неке карактеристичне тачке већа, угао мањи, пречник већи и слично).

Код примарног моделирања може да се специфицира један или више релационих исказа – тип “1” или тип “2”.

Специфицирање једног од ова два типа релационих исказа (тип “1”, тип “2”) омогућава да се елиминише један од два могућа скупа тачака који произилазе из осталог дела поступка моделирања ентитета.

Подтип типа “2” релационих исказа је “нешто” највеће или “нешто” најмање. Овим подтипом релационих исказа захтева се да “нешто” као што је вредност координате неке карактеристичне тачке или вредност неког геометријског параметра буде највећа или најмања (нпр.  $x$  координата централне тачке највећа, пречник највећи и слично).

Издвајање једног скупа тачака, односно двоструко мање од два, четири или осам ентитета који произилазе из осталог дела операција, поткоманди и слично, у поступку моделирања, и то у његовом првом делу, може да се изведе и специфицирање трећег релационог исказа (тип "3"). Ово је релациони исказ типа "нешто" даље од "нечега" или "нешто" ближе "нечему". Овде је "нешто" карактеристична тачка (почетна, крајња, средња или централна) ентитета који се моделира, а "нечега" или "нечему" је друга специфицирана тачка (најпогодније карактеристична тачка неког другог ентитета у сложеном ентитету који се моделира). Релације у релационом исказу тип "3" су ближе или даље. Овим типом релационих исказа захтева се да удаљеност неке карактеристичне тачке ентитета који се моделира буде већа од друге специфициране тачке ван ентитета који се моделира код релационих исказа "нешто" даље од "нечега". Овим типом релационих исказа се захтева да удаљеност неке карактеристичне тачке ентитета који се моделира буде мања од друге специфициране тачке ван ентитета који се моделира код релационих исказа "нешто" ближе "нечему".

Подтип типа "3" релационих исказа је "нешто" најдаље од "нечега" или "нешто" најближе "нечему" [39]. Овим подтипом релационих исказа се захтева да удаљеност неке карактеристичне тачке ентитета који се моделира од друге специфициране тачке буде највећа код релационих исказа "нешто" најдаље од "нечега". Овим подтипом релационих исказа захтева се да удаљеност неке карактеристичне тачке ентитета који се моделира од друге специфициране тачке буде најмања код релационих исказа "нешто" најближе "нечему".

Код поступка за моделирање (друга фаза) укључује се релациони исказ, врши се израчунавање израза (у исказу) ако је потребно, врши се испитивање истинитости (тачности) релационог исказа. Издвајање решења за ентитет који се моделира изводи се тако да се преузима (издваја) оно решење из којег подаци доводе до истинитости релационог исказа.

Код релационог исказа тип "3", а најчешће и код релационог исказа тип "2" и тип "1", није потребно израчунавање израза са једне стране релације (тип "2") или обе стране релације (тип "1" и тип "3"). Ово важи ако су са једне стране релације (тип "2") или обе стране релације (тип "1") дате само координате карактеристичних тачака или други карактеристични геометријски параметри.

Код релационог исказа тип "3" ради се о тачкама па овде и нема израза, изузев у случају да се уместо друге тачке неког другог ентитета која се налази у овом исказу специфицира посебна тачка у којој се координате добијају помоћу израчунавања израза. Овде се обавезно израчунава дистанца између тачака датих у релационом исказу.

У групи за додатни (секундарни) опис ентитета може да се да, на пример, тип линије у S5. Ту могу да буду наведени разни типови линија, као што су: пуна линија, црта-бланко, црта-бланко-тачка-бланко, итд.

На крају се може у овој групи дати и дебљина линије S6, као и боја у S7. Може се овде дати евентуално и страна (*layer*) у S8 на којој неки ентитет треба да се црта.

Овде могу да постоје стварни или фиктивни ентитети, тј. ентитети који се цртају и који се не цртају. Фиктивни ентитети се користе, пре свега, за лакши опис неког стварног ентитета. Фиктивни ентитети поседују све карактеристике као и стварни ентитети, и у односу на њих могу се користити релације положаја,

као и све друге њихове карактеристике (карактеристичне тачке, координате карактеристичних тачака и остале карактеристике ) за опис неког другог стварног или фиктивног ентитета.

Те карактеристике могу се позвати код описа директно или индиректно (подразумевају се).

Са једном датотеком (листом и сл.) у којој се налази опис групе основних ентитета (где су описи појединих основних ентитета јасно раздвојени) може се описати сваки сложени графички ентитет [46]:

(идентификација; ознака ентитета; примарни и секундарни опис)

(идентификација; ознака ентитета; примарни и секундарни опис)

...

(идентификација; ознака ентитета; примарни и секундарни опис)

Код неких линија-команди за опис појединог основног ентитета засебно, или уколико се линија-команда за опис неког ентитета налази у групи за опис неког сложеног ентитета, заједно са другим основним ентитетима, може постојати нека од следећих варијанти:

- да се унапред подразумева изостављени део за опис ентитета, тј. елемент или део елемента за опис ентитета,
- да се узима (подразумева) задње дефинисани део описа из неке претходне линије, тј. команде за опис неког другог ентитета за изостављени део у колекцији или неком делу колекције за опис ентитета,
- да се узимају неки делови описа из других ентитета (може се позвати било која карактеристика коју позвани ентитет поседује).

Сагласно овоме, изостављање неког елемента за опис из колекције или дела елемента из колекције за опис, у смислу да се он подразумева ако се понавља, да није потребан у овој или оној варијанти из опште структуре, да се додељује аутоматски и слично, не мења основну структуру и суштину језика.

У свим изразима за опис тачака, односно координата тачака или било којих других параметара, као и у свим релационим исказима и код дефинисања домена и коодомена када се ентитет описује функцијама, код примарног описа једног основног ентитета или у једном делу и код описа модификација (пресликавање, транслација и сл.), поред константи, променљивих (варијабли), аритметичких операција, степеновања и стандардних математичких функција, могу се користити координате и други геометријски параметри из осталих ентитета или истог ентитета, као и координате тачака пресека ентитета.

Иначе, као што је речено, могу се користити, тј. позвати код описа неког ентитета све карактеристике било ког другог дефинисаног ентитета. Притом је потребно навести ту карактеристику (карактеристичну тачку, координату карактеристичне тачке и друге параметре, као што су пречник, полупречник, угао, дужина итд.) или релацију положаја (тангентно, управно, паралелно итд.) и извршити селекцију ентитета од којег се та карактеристика узима. На исти начин се могу преузети и карактеристике из секундарног дела описа (боја, тип линије итд.).

Начин уноса и селекције (тастатура, миш итд.) елемената колекције или делова елемената колекције за опис једног ентитета, засебно или у групи са више основних ентитета, код моделирања, односно цртања једног или више основних ентитета или код генерисања скупа за опис (датотеке, листе и сл. за опис једног или више основних ентитета) не мења суштину овог језика.

Ни редослед уноса или селекције појединих елемената из колекције за опис (S1, S2, ..., Si, ..., S8) као ни делова неког елемента (Si) из колекције не мења суштину овог језика.

Исто тако, ни положај појединих елемената из колекције за опис (S1, S2, ..., Si, ..., S8) у оквиру колекције за опис једног ентитета, као ни сегмената за опис у оквиру неког елемента (Si), а такође ни положај описа једног ентитета у оквиру описа групе ентитета не мења суштину овог језика.

Напомене наведене на крају ове тачке важе и за све остале опште и посебне делове наведене у даљим објашњењима овог језика.

Коришћењем примарног дела описа ентитета могу се описати делимично модификације (нпр. пресликавање, транслација и ротација).

### **3.2. Елементи у структури примарног (геометријског) описа ентитета**

#### **Додељивање вредности карактеристичним тачкама ентитета (почетна тачка, крајња тачка, средња тачка, тачка центра)**

Додељивање вредности за уређене парове координата карактеристичних тачака може да се изведе на један од следећих начина:

Карактеристична тачка ← константа<sup>1</sup>, константа<sup>2</sup>

Карактеристична тачка ← промењива<sup>1</sup>, промењива<sup>2</sup>

Карактеристична тачка ←  $x$  или  $y$  у координата неке карактеристичне тачке другог ентитета<sup>1</sup>,  $x$  или  $y$  у координата неке карактеристичне тачке другог ентитета<sup>2</sup>

Карактеристична тачка ←  $x$  или  $y$  у координата тачке пресека два друга ентитета<sup>1</sup>,  $x$  или  $y$  у координата тачке пресека два друга ентитета<sup>2</sup>

Карактеристична тачка ← аритметички израз<sup>1</sup>, аритметички израз<sup>2</sup>

Могуће су и све комбинације 1 и 2 код дефинисања карактеристичних тачака, као на пример,

Карактеристична тачка ← константа, променљива

Карактеристична тачка ← променљива, аритметички израз итд.

Могуће је и директно додељивање вредности тачака из других ентитета:

Карактеристична тачка ← карактеристична тачка другог ентитета

Карактеристична тачка ← тачка пресека два ентитета

У аритметички израз могу да буду укључени елементи као што су: константе, променљиве, знакови за сабирање, знакови за одузимање, знакови за множење, знакови за дељење, знакови за степеновање, стандардне математичке функције, заграде,  $x$  или  $y$  у координате карактеристичних тачака,  $x$  или  $y$  у координате тачака пресека два ентитета, параметри других ентитета, као што су дужина, угао, пречник, полупречник, почетни и крајњи параметар домена из ентитета који се описују функцијама.

Сходно садржају и структури аритметичког изрази, додељивање вредности координатама у уређеном пару на начин како је дато у прва четири наведена случаја или у комбинацијама 1 и 2 из прва четири случаја, је само један посебан случај додељивања вредности аритметичких изрази.

### **Додељивање вредности координатама карактеристичних тачака ентитета (почетна тачка, крајња тачка, средња тачка, тачка центра)**

$x$  или  $y$  у координата карактеристичне тачке ← константа

$x$  или  $y$  у координата карактеристичне тачке ← променљива

$x$  или  $y$  у координата карактеристичне тачке ←  $x$  или  $y$  у координата неке карактеристичне тачке другог ентитета

$x$  или  $y$  у координата карактеристичне тачке ←  $x$  или  $y$  у координата тачке пресека два друга ентитета

$x$  или  $y$  у координата карактеристичне тачке ← аритметички израз

У аритметички израз могу да буду укључени елементи као што су константе, променљиве, знакови за сабирање, знакови за одузимање, знакови за множење, знакови за дељење, знакови за степеновање, стандардне математичке функције, заграде,  $x$  или  $y$  у координате карактеристичних тачака,  $x$  или  $y$  у координате тачака пресека два ентитета, параметри других ентитета, као што су дужина, угао, пречник, полупречник, почетни и крајњи параметар домена из ентитета који се описују функцијама.

Сходно садржају и структури аритметичког изрази, додељивање вредности координатама карактеристичних тачака засебно на начин како је дато у прва четири случаја само је посебан случај додељивања вредности аритметичког изрази.

### **Додељивање вредности карактеристичним параметрима (дужина, угао, пречник, полупречник) ентитета**

карактеристични параметар ← константа

карактеристични параметар ← променљива

карактеристични параметар ← карактеристични параметар другог ентитета

карактеристични параметар ← аритметички израз

У аритметички израз могу да буду укључени елементи као што су:

константе, променљиве, знакови за сабирање, знакови за одузимање, знакови за множење, знакови за дељење, знакови за степеновање, стандардне математичке функције, заграде,  $x$  или  $y$  у координате карактеристичних тачака,  $x$  или  $y$  у координате тачака пресека два ентитета, параметри других ентитета, као што су

дужина, угао, пречник, полупречник, почетни и крајњи параметар домена из ентитета који се описују функцијама.

Сходно садржају и структури аритметичког израза, додељивање вредности карактеристичним параметрима на начин како је дато у прва три случаја само је посебан случај додељивања вредности аритметичког израза.

### Релације положаја ентитета који се дефинише према другом ентитету

У следећој табели дате су релације између ентитета:

Табела 1: Релације између ентитета

Ентитет који се дефинише	Релација	Ентитет у односу на кога се дефинише
Дуж	паралелно	Дуж
Дуж	управно	Дуж, круг или кружни лук
Дуж	тангентно	Круг или кружни лук
Круг	тангентно	Дуж, круг или кружни лук
Кружни лук	тангентно	Дуж, круг или кружни лук

У опису се даје релација и одабрани ентитет (ентитет у односу на који се односи релација према ентитету који се дефинише).

У опису се може дати и ентитет који се дефинише уз релацију и одабрани ентитет (ентитет у односу на који се односи релација према ентитету који се дефинише).

### Релациони искази [56,60,63]

Овде постоје три типа релационих исказа и два подтипа релационих исказа.

Тип 1:

Општа дефиниција:

“Нешто” мање од “Нечега”

“Нешто” веће од “Нечега”

Овде се даје структура израза са леве стране релације “мање од” или релације “веће од”:

Нешто  $\leftarrow$   $x$  или  $y$  у координата неке карактеристичне тачке ентитета који се дефинише

Нешто  $\leftarrow$  карактеристични параметар ентитета који се дефинише

Нешто  $\leftarrow$   $x$  или  $y$  у координата тачке пресека ентитета који се дефинише и другог ентитета

Нешто  $\leftarrow$  аритметички израз

У аритметички израз могу да буду укључени елементи као што су:

константе, променљиве, знакови за сабирање, знакови за одузимање, знакови за множење, знакови за дељење, знакови за степеновање, стандардне математичке функције, заграде,  $x$  или  $y$  у координате карактеристичних тачака,  $x$  или  $y$  у координате тачака пресека два ентитета, параметри других ентитета, као што су

дужина, угао, пречник, полупречник, почетни и крајњи параметар домена из ентитета који се описују функцијама.

Сходно садржају и структури аритметичког израза, додељивање вредности за “Нешто” на начин како је дато у прва три случаја само је посебан случај четвртог начина додељивања вредности за “Нешто” у релационом исказу.

Овде се даје структура израза са десне стране релације “мање од” или релације “веће од”:

Нечега ← константа

Нечега ← променљива

Нечега ←  $x$  или  $y$  у координата неке карактеристичне тачке ентитета који се дефинише или другог ентитета

Нечега ←  $x$  или  $y$  у координата тачке пресека ентитета који се дефинише и другог ентитета (ознака координате тачке пресека и идентификација ентитета који се секу)

Нечега ← карактеристични параметар ентитета који се дефинише

Нечега ← аритметички израз

У аритметички израз могу да буду укључени елементи као што су:

константе, променљиве, знакови за сабирање, знакови за одузимање, знакови за множење, знакови за дељење, знакови за степеновање, стандардне математичке функције, заграде,  $x$  или  $y$  у координате карактеристичних тачака,  $x$  или  $y$  у координате тачака пресека два ентитета, параметри других ентитета, као што су дужина, угао, пречник, полупречник, почетни и крајњи параметар домена из ентитета који се описују функцијама.

Сходно садржају и структури аритметичког израза, додељивање вредности за “Нечега” на начин како је дато у првих пет случајева само је посебан случај шестог начина додељивања вредности за “Нечега” у релационом исказу.

Тип 2:

Општа дефиниција:

“Нешто” веће

“Нешто” мање

Може и “веће Нешто”, односно “мање Нешто”.

Овде се даје структура израза “Нешто”:

Нешто ←  $x$  или  $y$  у координата неке карактеристичне тачке ентитета који се дефинише

Нешто ← карактеристични параметар ентитета који се дефинише

Нешто ←  $x$  или  $y$  у координата тачке пресека ентитета који се дефинише и другог ентитета

Нешто ← аритметички израз

У аритметички израз могу да буду укључени елементи као што су:

константе, променљиве, знакови за сабирање, знакови за одузимање, знакови за множење, знакови за дељење, знакови за степеновање, стандардне математичке функције, заграде,  $x$  или  $y$  у координате карактеристичних тачака,  $x$  или  $y$

координате тачака пресека два ентитета, параметри других ентитета као што су дужина, угао, пречник, полупречник, почетни и крајњи параметар домена из ентитета који се описују функцијама.

Сходно садржају и структури аритметичког израза, додељивање вредности за “Нешто” на начин како је дато у прва три случаја само је посебан случај четвртог начина додељивања вредности за “Нечега” у релационом исказу.

Подтип типа 2:

Врши се спецификација исказа

“Нешто” највеће

или

“Нешто” најмање

или другачије постављеног исказа, као на пример: највеће “Нешто”, односно најмање “Нешто”.

Овде се даје структура израза “Нешто”:

Нешто  $\leftarrow$   $x$  или  $y$  у координата неке карактеристичне тачке ентитета који се моделира

Нешто  $\leftarrow$  карактеристични параметар ентитета који се моделира

Нешто  $\leftarrow$  дистанца између карактеристичне тачке ентитета који се моделира и друге специфициране тачке.

Тип 3:

Врши се спецификација исказа

“Нешто” даље од “Нечега”

или

“Нешто” ближе “Нечему”

Овде се даје структура израза са леве и десне стране релације “даље од” или релације “ближе”:

Нешто  $\leftarrow$  нека карактеристична тачка ентитета који се моделира

Нечега  $\leftarrow$  тачка ван ентитета који се моделира (најпогодније карактеристична тачка неког другог ентитета)

Нечему  $\leftarrow$  тачка ван ентитета који се моделира (најпогодније карактеристична тачка неког другог ентитета)

Нечега  $\leftarrow$  израз за  $x$  координату тачке, израз за  $y$  координату тачке

Нечему  $\leftarrow$  израз за  $x$  координату тачке, израз за  $y$  координату тачке

У израз за  $x$  координату или  $y$  координату могу да буду укључени елементи и подоперације као што су: константе, променљиве, сабирање, одузимање, множење, дељење, степеновање, стандардне математичке функције,  $x$  или  $y$  координате карактеристичних тачака,  $x$  или  $y$  координате тачака пресека два ентитета, параметри других ентитета, као што су дужина, угао, пречник,



полупречник, почетни и крајњи параметар домена из ентитета који се моделирају функцијама.

Подтип типа 3:

Врши се спецификација исказа

“Нешто” најдаље од “Нечега”

или

“Нешто” најближе “Нечему”

Овде се даје структура израза са леве и десне стране релације “најдаље од” или релације “најближе”:

Нешто ← нека карактеристична тачка ентитета који се моделира

Нечега ← тачка ван ентитета који се моделира (најпогодније карактеристична тачка неког другог ентитета)

Нечему ← тачка ван ентитета који се моделира (најпогодније карактеристична тачка неког другог ентитета)

Нечега ← израз за  $x$  координату тачке, израз за  $y$  координату тачке

Нечему ← израз за  $x$  координату тачке, израз за  $y$  координату тачке

У израз за  $x$  координату или  $y$  координату могу да буду укључени елементи и подоперације као што су: константе, променљиве, сабирање, одузимање, множење, дељење, степеновање, стандардне математичке функције,  $x$  или  $y$  координате карактеристичних тачака,  $x$  или  $y$  координате тачака пресека два ентитета, параметри других ентитета, као што су дужина, угао, пречник, полупречник, почетни и крајњи параметар домена из ентитета који се моделирају функцијама.

**Домен (почетна и крајња вредност) у којем се израчунава функција задата изразом код ентитета који се описују функцијама**

Додељивање вредности домену код описа ентитета функцијама у непараметарском облику:

Почетна или крајња вредност домена (за  $x$  или  $y$  координату домена) ← константа

Почетна или крајња вредност домена (за  $x$  или  $y$  координату домена) ← променљива

Почетна или крајња вредност домена (за  $x$  или  $y$  координату домена) ←  $x$  или  $y$  координата неке карактеристичне тачке другог ентитета

Почетна или крајња вредност домена (за  $x$  или  $y$  координату домена) ←  $x$  или  $y$  координата тачке пресека друга два ентитета

Почетна или крајња вредност домена (за  $x$  или  $y$  координату домена) ← аритметички израз

У аритметички израз могу да буду укључени елементи као што су константе, променљиве, знакови за сабирање, знакови за одузимање, знакови за множење, знакови за дељење, знакови за степеновање, стандардне математичке функције, заграде,  $x$  или  $y$  координате карактеристичних тачака,  $x$  или  $y$  координате тачака

пресека два ентитета, параметри других ентитета, као што су дужина, угао, пречник, полупречник, почетни и крајњи параметар домена из ентитета који се описују функцијама.

Сходно садржају и структури аритметичког израза, додељивање вредности почетној и крајњој координати домена на начин како је дато у прва четири случаја само је посебан случај петог начина додељивања вредности за домен.

Додељивање вредности домену код описа ентитета функцијама у параметарском облику:

Почетна или крајња вредност домена (за параметар домена) ← константа

Почетна или крајња вредност домена (за параметар домена) ← променљива

Почетна или крајња вредност домена (за параметар домена) ← почетна или крајња вредност параметра домена из другог ентитета

Почетна или крајња вредност домена (за параметар домена) ← аритметички израз

У аритметички израз могу да буду укључени елементи као што су константе, променљиве, знакови за сабирање, знакови за одузимање, знакови за множење, знакови за дељење, знакови за степеновање, стандардне математичке функције, заграде,  $x$  или  $y$  у координате карактеристичних тачака,  $x$  или  $y$  у координате тачака пресека два ентитета, параметри других ентитета, као што су дужина, угао, пречник, полупречник, почетни и крајњи параметар домена из ентитета који се описују функцијама.

Сходно садржају и структури аритметичког израза, додељивање вредности почетном и крајњем параметру домена на начин како је дато у прва три случаја је само посебан случај четвртог начина додељивања вредности за домен.

### **Кодомен по којем се израчунава координата задата изразом код ентитета који се описују функцијама**

Структура израза за кодомен код описа ентитета функцијама у непараметарском облику:

Израз за  $y$  или  $x$  координату кодомена ← константа

Израз за  $y$  или  $x$  координату кодомена ← променљива

Израз када је  $y$  координата кодомен ← аритметички израз<sup>1</sup>

Израз када је  $x$  координата кодомен ← аритметички израз<sup>2</sup>

У аритметички израз<sup>1</sup> могу да буду укључени елементи као што су константе, променљиве, знакови за сабирање, знакови за одузимање, знакови за множење, знакови за дељење, знакови за степеновање, стандардне математичке функције, заграде,  $x$  или  $y$  у координате карактеристичних тачака,  $x$  или  $y$  у координате тачака пресека два ентитета, параметри других ентитета, као што су дужина, угао, пречник, полупречник,  $x$  координата (координата домена) из ентитета који се тренутно описује функцијама.

У аритметички израз<sup>2</sup> могу да буду укључени елементи као што су константе, променљиве, знакови за сабирање, знакови за одузимање, знакови за множење, знакови за дељење, знакови за степеновање, стандардне математичке функције,

заграде,  $x$  или  $y$  координате карактеристичних тачака,  $x$  или  $y$  координате тачака пресека два ентитета, параметри других ентитета, као што су дужина, угао, пречник, полупречник,  $y$  координата (координата домена) из ентитета који се тренутно описује функцијама.

Сходно садржају и структури аритметичког израза<sup>1</sup> или аритметичког израза<sup>2</sup> за кодомен, прва два случаја израза за кодомен су само посебни случајеви аритметичког израза.

Структура израза за кодомен код описа ентитета функцијама у параметарском облику:

Израз за  $y$  и  $x$  координату кодомена  $\leftarrow$  константа

Израз за  $y$  и  $x$  координату кодомена  $\leftarrow$  промењива

Израз за  $y$  и  $x$  координату кодомена  $\leftarrow$  аритметички израз

У аритметички израз могу да буду укључени елементи као што су: константе, променљиве, знакови за сабирање, знакови за одузимање, знакови за множење, знакови за дељење, знакови за степеновање, стандардне математичке функције, заграде,  $x$  или  $y$  координате карактеристичних тачака,  $x$  или  $y$  координате тачака пресека два ентитета, параметри других ентитета, као што су дужина, угао, пречник, полупречник, параметар домена код ентитета који се описују функцијама.

Сходно садржају и структури аритметичког израза за кодомен, прва два случаја додељивања вредности за кодомен су само посебни случајеви аритметичког израза.

### 3.3. Дефиниције примарног (геометријског) описа ентитета

Овде ће се дати само један број дефиниција дужи ради даљњег разјашњења опште структуре језика за моделирање. Део дефиниција дужи и осталих ентитета је дат у посебној тачки, у којој су дате графичка представа ентитета који се моделира, а поред ње општа дефиниција и примери дефинисања ентитета.

Ентитети *дуж*, *кружни лук* и *крива линија* су оријентисани од почетне тачке ка крајњој тачки.

#### Оријентисана дуж

Табела 2: Оријентисана дуж I

Тачка		Параметар		Релација			Релациони исказ	
почетна	крајња	дужина	угао	пар.	упр.	тан.	тип1	тип2
X	X							
X		X	X					
X		X		X			X	
X		X		X				X

Напомена 1: Угао дужи се даје у односу на  $x$ -осу (позитиван смер).

Напомена 2: Скраћенице *пар.*, *упр.*, *тан.* означавају редом: *паралелно*, *управно*, *тангентно*. Важе у свим табелама.

Табела 3: Оријентисана дуж II

Тачка		Параметар		Релација			Релациони исказ	
почетна	средња	дужина	угао	пар.	упр.	тан.	тип1	тип2
X	X							
	X	X	X					
	X	X		X			X	
	X	X		X				X

Напомена: Угао дужи се даје у односу на *x*-осу (позитиван смер).

Табела 4: Оријентисана дуж III

Тачка		Параметар		Релација			Релациони исказ	
почетна	крајња	дужина	угао	пар.	упр.	тан.	тип1	тип2
X		X			X		X	
X		X			X			X
	X	X			X		X	
	X	X			X			X

Напомена: Ако се изостави *параметар дужина* и *релациони исказ*, подразумева се да друга тачка (почетна или крајња, која није наведена у дефиницији) лежи на ентитету у односу на који је управна дуж која се дефинише. Дефиниција оријентисане дужи је тада као у следећој табели:

Табела 5: Оријентисана дуж IV

Тачка		Параметар		Релација			Релациони исказ	
почетна	крајња	дужина	угао	пар.	упр.	тан.	тип1	тип2
X					X			
	X				X			

Може да се да и средња тачка као у дефиницијама у следећој табели:

Табела 6: Оријентисана дуж V

Тачка	Параметар		Релација			Релациони исказ	
средња	дужина	угао	пар.	упр.	тан.	тип1	тип2
X	X			X		X	
X	X			X			X

Табела 7: Оријентисана дуж VI

Тачка		Параметар		Релација			Релациони исказ	
почетна	крајња	дужина	угао	пар.	упр.	тан.	тип1	тип2
X		X				X	X	
X		X				X		X

Напомене: Овде се подразумева да је дуж оријентисана ка ентитету у односу на који је тангентна дуж која се дефинише: довољан је један *релациони исказ*.

Могу се дати и дефиниције где се не подразумева унапред оријентација дужи. Тада је потребно дати два *релациона исказа*.

Ако се изостави *параметар дужина*, подразумева се да друга тачка (почетна или крајња, која није наведена у дефиницији) лежи на ентитету у односу на који је тангентна дуж која се дефинише.

Дефиниција оријентисане дужи је тада као у следећој табели:

Табела 8: Оријентисана дуж VII

Тачка		Параметар		Релација			Релациони исказ	
почетна	крајња	дужина	угао	пар.	упр.	тан.	тип1	тип2
X						X	X	
X						X		X

Табела 9: Оријентисана дуж VIII

Тачка почетна	x-коорд. тачке крајње	y-коорд. тачке крајње	Параметар		Релација			Релациони исказ	
			дужина	угао	пар.	упр.	тан.	тип1	тип2
X	X			X					
X		X		X					
X	X		X					X	
X		X	X					X	
X	X		X						X
X		X	X						X

Напомена: Угао дужи се даје у односу на x-осу (позитиван смер).

Табела 10: Оријентисана дуж IX

Тачка почетна	x-коорд. тачке крајње	y-коорд. тачке крајње	Параметар		Релација			Релациони исказ	
			дужина	угао	пар.	упр.	тан.	тип1	тип2
X	X				X				
X		X			X				

Табела 11: Оријентисана дуж X

Тачка крајња	x-коорд. тачке почетне	y-коорд. тачке почетне	Параметар		Релација			Релациони исказ	
			дужина	угао	пар.	упр.	тан.	тип1	тип2
X	X				X				
X		X			X				

Табела 12: Оријентисана дуж XI

Тачка средња	x-коорд. тачке почетне	y-коорд. тачке почетне	Параметар		Релација			Релациони исказ	
			дужина	угао	пар.	упр.	тан.	тип1	тип2
X	X				X				
X		X			X				

Табела 13: Оријентисана дуж XII

Тачка средња	x-коорд. тачке крајње	y-коорд. тачке крајње	Параметар		Релација			Релациони исказ	
			дужина	угао	пар.	упр.	тан.	тип1	тип2
X	X				X				
X		X			X				

Дефиниције које произилазе из опште структуре и елемената у структури примарног (геометријског) описа, са задатом крајњом или средњом тачком, имају сличну структуру као када се задаје почетна тачка (као што је у дефиницијама датим овде, у табелама XI и XII) па се даље неће понављати.

Уместо тачака као уређених парова у дефиницијама могу да стоје координате тачака засебно, као у дефиницијама датим у следећим табелама.

Табела 14: Оријентисана дуж XIII

Тачка почетна		x-коорд. тачке крајње	y-коорд. тачке крајње	Параметар		Релација			Релациони исказ	
x-коорд.	y-коорд.			дужина	угао	пар.	упр.	тан.	тип1	тип2
X	X	X				X				
X	X		X			X				

Из опште структуре и елемената у структури примарног (геометријског) описа, као и ових примера дефиниција, очигледне су остале дефиниције оријентисане дужи, па се неће даље приказивати табеларно остале дефиниције.

Овде се могу поменути само неке специфичне дефиниције где постоје две *релације*. Једна од таквих дефиниција је, на пример, дуж тангентна на један лук или кружницу, а истовремено управна на неку дуж.

Дефиниција овакве дужи:

Релација тангентно на један лук или кружницу; релација управно на једну дуж; релациони исказ; релациони исказ.

Једним релационим исказом одређује се која се од две дужи издваја, а другим релационим исказом одређује се оријентација дужи.

У дефиницијама где постоји релација *дуж тангентна на један лук или кружницу*, а истовремено релација *дуж управна на другу дуж* могуће су комбинације са x или у координатом почетне тачке и x или у координатом крајње тачке. Овде је потребан још један *релациони исказ* (у дефиницијама) за издвајање

једне од две дужи које настају као резултат ових дефиниција без *релационог исказа*.

Из опште структуре и елемената у структури примарног (геометријског) описа, као и ових примера дефиниција, очигледне су и дефиниције ентитета кружни лук и круг (свуда у тексту овог рада под речју *круг* подразумева се *кружница*), па се неће овде даље табеларно приказивати ни објашњавати дефиниције тих ентитета.

Овде је дат само један део од мноштва дефиниција које произилазе из опште структуре и елемената у структури примарног (геометријског) описа, у намери да се прикаже како се у принципу користе елементи у структури примарног (геометријског) описа.

Дефиниције из претходних табела, у којима су укључени релациони искази тип 1 или тип 2, у опису ентитета оријентисана дуж, могу бити и са релационим исказом тип 3.

## 4. СТРУКТУРА ЈЕЗИКА ЗА ПАРАМЕТАРСКО ГЕОМЕТРИЈСКО МОДЕЛИРАЊЕ 2/3D

Језик за моделирање и цртање биће описан на примеру језика за линијско 2D и делимично 3D моделирање и цртање.

За детаљније објашњење овог језика усваја се, као што је дато у наредном делу текста, одређена лексичка, семантичка и синтаксичка структура за 2D моделирање и цртање и 3D моделирање и цртање (линијски, тј. жичани опис).

### Лексичка структура

Алфабет језика чине следећи скупови знакова:

- велика и мала слова енглеског алфабета: A, B, ... , Z; a, b, ... , z;
- знаменке: 0, 1, 2, ..., 9;
- остали знакови : { [ ] } ( ) ; : , = % ~ + - \* / < > \_ . и бланко.

Једнако значење има употреба великих и малих слова алфабета.

Речник чине следеће класе речи:

- константе (цели и реални бројеви и специјална константа  $\pi$ );
- кључне речи: L, O, C, F, WL, WO, WC, WF, MM, MLM, M, N, C, T, S, XM, XN, XC, XT, XP, XS, YM, YN, YC, YT, YP, YS, ZM, ZN, ZC, ZT, ZP, ZS, R, D, L, A, AM, AN (2D);
- стандардне математичке функције као што су: sin, cos, tg, ctg, arctg, log, ln, итд;
- идентификатори: Имена променљивих могу да се састоје од слова и знаменки, при чему на првом месту мора да буде слово.
- остале речи: овде се подразумевају речи које не припадају ниједној од горе наведених класа, а то су: релације >, <, >=, <= и једнакост =.

Операције: +, -, \*, /, ~ (\* – ознака за множење), (/ – ознака за дељење), (~ – ознака за степеновање), (% – ознака за степене код углова), специјалне функције:

T, P, S, M, N, C, R, D, L, XM, YM, ZM, XN, YN, ZN, XC, YC, ZC, XP, YP, ZP, XS, YS, ZS, XT, YT, ZT, XР, YР, ZР, A, AM, AN (углови 2D)

- релације положаја // (паралелно), IC (тангентно), I- (управно)
- тип линије \_ , \_\_ , \_: , \_: , ... , \_: , \_: , и остали типови линија од комбинација црта, тачка, бланко, и остале речи, као што су маркери и слично.

У целом тексту *круг* је *кружница*, а *лук* ако није наглашено – *кружни лук*.



## Опис ентитета – семантика језика и синтаксичка структура

Сваки основни графички ентитет описује се скупом који се састоји од подскупова, односно елемената за опис ентитета. Колекција елемената за опис ентитета  $S$  може се дати у општем случају као:

$$S = \{S_1, S_2, S_3, S_4, S_5\}$$

односно може се, као што је наведено у општој структури, ова колекција проширити са  $S_6$  и  $S_7$ . Почетни део описа јесте нека идентификација или број ентитета  $S_1 \in \{1, 2, 3, \dots\}$ . Број ентитета је идентификатор основног ентитета у оквиру једне позиције, пројекције, погледа или детаља. Овај елемент колекције  $S$  за опис ентитета потребан је због тога што је најчешће погодније за геометријски опис позвати се на неки елемент скупа раније дефинисаног ентитета него давати изнова одговарајуће параметре.

Семантика језика је тако овде одабрана да, где је то могуће, асоцира на значење или се користе уобичајени симболи из математике. Тако је, на пример, скуп симбола за опис врсте ентитета  $S_2 \in \{L, O, C, F\}$ .

Елемент из овог скупа симбола одређује врсту ентитета дуж, круг, кружни лук или другу криву линију. Овде симболи асоцирају на врсту ентитета јер су слични геометрији ентитета које означавају, изузев симбола  $F$ , који је уобичајен за функцију.

Уколико се ради о фиктивном ентитету, додаје се испред ових симбола симбол  $W$ .

Следећи из скупа, којим се формира ентитет, постоји само код ентитета који се дефинише функцијом. Ако је други елемент из скупа елемената за опис ентитета задњи из скупа  $S_2$ , тј.  $F$  тада постоји и функција којом се одређује кодомен функције на задатом домену. На пример,  $S_3$  је једна или две функције облика  $y=f(x)$  или  $x=f(y)$ , односно у параметарском облику  $y=f_1(t)$ ,  $x=f_2(t)$  код  $2D$  моделирања.

Сваком елементу  $x \in X$  ( $X$  је подскуп реалних бројева) придружује се тачно један елемент  $y=f(x) \in Y$  по неком правилу  $F$  за функцију типа  $y=f(x)$ , односно сваком елементу из области дефинисаности придружује се једна вредност из скупа вредности.

$S_3$  је празан скуп ако је  $S_2 \in \{L, O, C\}$ , односно ако је ентитет дуж, круг или кружни лук.

Даље долази део групе елемената за дефинисање геометрије у  $S_4$ , а то може да буде скуп тачака, координата тачака и других параметара, као што су углови и карактеристична растојања:

$$S, M, N, C, R, D, L, A, AM, AN, x_m, y_m, z_m, x_n, y_n, z_n, x_c, y_c, z_c, x_s, y_s, z_s$$

где су карактеристичне тачке подскупови координата

$$M = \{x_m, y_m\}$$

$$M = \{x_m, y_m, z_m\}$$

$$N = \{x_n, y_n\}$$

$$N = \{x_n, y_n, z_n\} \quad - \text{ у простору.}$$

$$C = \{x_c, y_c\}$$

$$C = \{x_c, y_c, z_c\}$$

$$S = \{x_s, y_s\}$$

$$S = \{x_s, y_s, z_s\}.$$

Дакле,  $S_4$  је скуп параметара којима се ближе описује геометрија датог ентитета.

Овде колекција параметара, односно елемената скупа  $S_4$  зависи од врсте ентитета.

Следећим елементом, односно скупом од једног елемента  $S_5$  даје се тип линије. На пример, колекција симбола за типове линија најчешће коришћених у машинству је  $\{ \_ , \_ \_ , \_ \cdot \_ \}$ , односно  $S_5 \in \{ \_ , \_ \_ , \_ \cdot \_ \}$ .

Овим би био скоро дефинисан скуп  $S$ , и у једној од великог броја варијација могао би овако да изгледа за опис дужи у простору:

$$1;L: M=a,b,c;N=e,f,g; \_ \_$$

или за опис криве линије у равни

$$1;F:y=x^2-(1/2)(e*\cos(b*c)); x_m=a;x_n=b; \_ .$$

И на крају, у секундарном опису, као задњи подскуп из скупа  $S$ , може се дати, за потпуно дефинисање ентитета, дебљина линије и боја линије. На пример, уобичајени скуп дебљина које се користе у машинству из ког се узима  $S_6$  је

$$S_6 \in \{0.1,0.2,0.3,0.4,0.5\} .$$

Даље, груписањем низа једноставних ентитета могу се добити шире структуре. Група основних ентитета чини сложени ентитет.

Елементи скупа  $S_4$ , којим се дефинише геометрија основних ентитета, могу да буду карактеристичне тачке, подскупови координата карактеристичних тачака, координате карактеристичних тачака или други карактеристични геометријски параметри. Карактеристичне тачке су: почетна тачка  $M$ , крајња тачка  $N$  и центар  $C$  кружног лука (у даљем тексту лука) или кружнице (у даљем тексту круга) и средња тачка  $S$  дужи или кружног лука.

Може се користити и тачка пресека два ентитета ( $P$ ), као и координате ове тачке као карактеристика чија се вредност додељује или се користи у исказима (изразима).

Наведене тачке су уређени подскупови координата  $\{x,y,z\}$ , односно у равни подскупови  $\{x,y\}$ . Могу се координате карактеристичних тачака користити не само као елементи поскупа тачака него и засебно.

Поред тачака и њихових координата, у овом језику могући елементи скупа за дефинисање појединих ентитета су дужина  $L$ , полупречник  $R$  и пречник круга  $D$ , угао дужи  $A$ , као и углови које захватају дуж  $CM$ , односно дуж  $CN$  кружног лука са позитивним делом  $x$ -осе (код  $2D$  моделирања).

Могу се дати и углови пројекција дужи у равнима координатног система  $XYZ$ , при чему прва два знака иза ознаке угла означавају раван, а задњи у односу на коју осу (позитиван смер) се мери угао. На пример,  $auxx$  означава угао пројекције дужи у равни  $ux$ , и то  $A$  у односу на  $x$ -осу (у складу са овим су и ознаке углова за остале пројекције дужи као ентитета и дужи  $CM$  и  $CN$ , које спајају центар лука са почетном, односно крајњом тачком).

Притом, и код моделирања у равни може да се у ознаци назначи у односу на коју осу се мери угао. На пример,  $Ax$  је угао дужи у односу на  $x$ -осу (има исто значење

као и угао  $A$  код  $2D$  моделирања);  $A_y$  је угао дужи у односу на  $y$ -осу (позитиван смер), итд.

Овим параметрима се дефинише скуп координата друге карактеристичне тачке (недефинисане тачке), односно одређује се у потпуности скуп тачака основног ентитета.

Ентитети (дуж, лук, круг) могу да буду дефинисани карактеристичним тачкама  $M, N, C$  код дужи, односно тачкама  $M, N, C, S$  код лука, и делимично са тачком  $C$  код круга.

У опису могу да се користе и координате карактеристичних тачака одвојено или у оквиру подскупа координата за неку карактеристичну тачку.

Поред тога се, ако је потребно, може комбиновати скуп  $S_4$  и са другим карактеристикама. Ту су не само угао и дужина код дефинисања дужи, дужина тетиве, полупречник и угао код дефинисања лука или пречник и полупречник код дефинисања круга итд. него и елементи скупа  $S_4$  код којих се полази од положаја ентитета који се дефинише према раније дефинисаном ентитету (тангентно, управно, паралелно).

Овде се у делу колекције за опис ентитета, у  $S_4$ , могу дати и релациони искази, којима се, ако то није дефинисано преосталим делом описа у  $S_4$ , прецизно (потпуно) дефинише скуп тачака ентитета једног основног ентитета. Вредности за поједине координате карактеристичних тачака и друге параметре ( $R, D, L, A, AM, AN, \dots$ ), којим се дефинишу ентитети, могу се дати у општем случају као аритметички изрази.

За елементе скупа којим је одређен поједини ентитет (дуж, круг, лук или крива линија) значење је уобичајено, па се било која координата, или неки од ових параметара као елементи скупа  $S_4$  могу уопште дати као

$$S_4i = a_i$$

што значи: израчунај па придружи вредност аритметичког израза  $a_i$  елементу скупа за дефинисање геометрије  $i$ -тог ентитета. На овај начин се елементима скупа  $S_4$  додељује вредност. Наводе се примери додељивања вредности елементима скупа којима се дефинишу основни ентитети

$$\begin{aligned} x_m &= a - b \cdot 20, \\ y_m &= q \cdot \cos(ax + bx), \\ z_m &= e \cdot x^2 - 2, \\ a &= \arctg(mx + d \cdot 2.0), \\ r &= x^2 + w \cdot \sin(\text{alfa} + \pi) \\ d &= (p^2 + 4)/2, \\ l &= ((q + e) \cdot 5 + c^2) \cdot \text{tg}(ax + \pi/2), \\ x_n &= 20, \\ y_n &= d \cdot 1 + 30, \\ z_n &= 50.20, \\ x_c &= a \cdot 1, \\ y_c &= (ac + ac1) \cdot 2, \\ z_c &= 1.5 \cdot qx. \end{aligned}$$

На сличан начин, могуће је и елементима подскупа карактеристичних тачака  $M = \{x_m, y_m, z_m\}$ ,  $N = \{x_n, y_n, z_n\}$ ,  $C = \{x_c, y_c, z_c\}$ ,  $S = \{x_s, y_s, z_s\}$  доделити вредност.

Почетна тачка, крајња тачка, центар лука или круга и средња тачка дужи или лука су подскупови или, прецизније, уређени парови за раван или тројке за дефинисање ентитета у простору. Може се, у овом смислу, дакле, свакој тачки из скупа тачака  $\{M, N, C, S\}$  као подсупу доделити вредност. Тачка  $T$  се овде уводи код описа ентитета када се користи релација (односно положаја између ентитета) тангентно.

Низ елемената скупа за тачку  $M$ , а тако је и за друге карактеристичне тачке, може се дати уопште на следећи начин:

$$k_i = a_m \varepsilon (x_i, y_i, z_i), \text{ где су } x_i, y_i, z_i \text{ координате тачке.}$$

На пример, ако се напише

$$M = a + b, c - d, (e * f + g * \sin(e1)) * 2$$

ово ће имати значење: израчунај па придружи редом вредности аритметичких израза и то:

$$\begin{aligned} X_M &\leftarrow a + b \\ Y_M &\leftarrow c - d \\ Z_M &\leftarrow (e * f + g * \sin(e1)) * 2. \end{aligned}$$

Дакле, свакој координати из уређеног пара или тројке придружује се вредност аритметичког израза по редоследу.

У аритметичком изразу могу се користити и карактеристични параметри других ентитета из неког сложеног ентитета. Ако се моделира, односно црта један објект или његова пројекција, поглед, детаљ итд., тада се могу за израчунавање и придруживање или само придруживање вредности координатама и параметрима као елементима скупа  $S_4$  користити вредности координата и свих других геометријских параметара других основних ентитета. Наравно, овде се могу користити само оне координате и параметри које „позвани” ентитет има. Тако се може написати и следеће

$$R = r(5)$$

што значи да полупречник  $R$  у ентитету који се описује узима вредност пречника из ентитета број 5. Ради илустрације даје се још неколико примера где се дефинишу вредности за координате карактеристичних тачака и друге параметре потребне за цртање основних ентитета, коришћењем параметара других ентитета:

$$\begin{aligned} L &= d(6) - m * \sin(gx) \\ R &= r(2) / 2 \\ x_m &= 3.0 + x_n(2) \\ a &= a(6) + \arcsin(q/ac) \\ y_c &= ab + y_m(5) \end{aligned}$$

$$D=(x_m(6)-x_c(6))*2.$$

За дефинисање уређених парова или тројки, односно скупа координата појединих тачака могу се користити поједине карактеристичне тачке ентитета који су раније дефинисани. Притом се за додељивање вредности не могу користити било какви аритметички изрази са десне стране знака једнакости, већ се ту врши додељивање целог подскупа координата. Могу се описати тачке на начин као што је дато у следећим примерима

$$M=M(5)$$

$$M=x_m(5),y_m(5)$$

$$M=x_m(5),y_m(5),z_m(5)$$

$$C=N(6)$$

$$C=x_n(6),y_n(6)$$

$$C=x_n(6),y_n(6),z_n(6)$$

$$N=M(8)$$

$$N=x_m(8),y_m(8)$$

$$N=x_m(8),y_m(8),z_m(8).$$

За дефинисање уређених парова или тројки може се користити и следећи начин додељивања вредности за поједине карактеристичне тачке:

$$M=x_n(1)+5.0,b+x_m(2),z_n(2)-L(7)$$

што значи да се додељује вредност почетној тачки тако да се одмах у опису вредности за ову тачку додељивање врши тако да се свакој координати додељује вредност одговарајућег израза, а тај израз може да се формира на исти начин као и код израза за додељивање вредности било којој координати засебно или било ком другом параметру.

Може се, код дефинисања елемената скупа  $S_4$  позвати и на раније дефинисани елемент овог скупа, из ентитета који се описује као у следећем примеру

$$12;C: M=2,2;N=x_m(12)+20.0,y_m(12)+30.0;A=\pi/2.$$

Придруживање целог подскупа координата једне тачке  $M$ ,  $N$ ,  $C$  или  $S$  некој другој у оквиру истог ентитета нема много смисла.

Релациони искази одређују ближе скуп тачака ентитета који се дефинише. Изрази са обе стране релација ( $<$ ,  $<=$ ,  $>$ ,  $>=$ ) или са једне стране код једног од могућих типова релационих исказа могу се дати на исти начин као и код дефинисања било које координате неке карактеристичне тачке или других параметара којима се описује неки ентитет. Овде се дају примери релационих исказа којима се од двају скупова тачака, који произилазе из осталог дела описа у  $S_4$ , издваја један (жељени) скуп:

$$x_m(12)>x_n(12) \text{ (ili } x_m>x_n)$$

$$z_m(18)+b*\cos(\beta)>z_n(15)$$

$$A(11)>\pi/4$$

$$x_m(15)< \text{ (ili } x_m<)$$

$$x_n(15)> \text{ (ili } x_n>).$$

Значење првог релационог исказа од ових пет примера је да се узме такав ентитет да  $x$ -координата почетне тачке ентитета са идентификацијом 12 буде већа од  $x$ -координате крајње тачке истог ентитета.

Значење два последња релациона исказа је захтев да се издвоји такав скуп тачака за неки ентитет, од два могућа, да се узме скуп тачака (ентитет 15) са мањом  $x_m$  координатом код претпоследњег примера, односно код последњег исказа захтева се да се узме скуп тачака, ентитета у којем се налази овај захтев, са већом  $x$ -координатом тачке N.

Користећи моделирање дефинисањем скупа S3 и S4, односно функцијско цртање, може се описати и у простору свака крива линија. Уређене тројке  $\{x,y,z\}$ , из скупа тачака дате криве, притом се добијају на аналоган начин као и у равни. Просторне криве се могу и овде дефинисати у непараметарском облику, експлицитно скупом једначина S3

$$x=x; y=f_1(x); z=f_2(x)$$

$$\text{или } x=f_1(y); y=y; z=f_2(y)$$

$$\text{или } x=f_1(z); y=f_2(z); z=z.$$

У скупу S3 може се дати и конкретан параметарски опис кривих чији је општи облик

$$x=x(t), y=y(t), z=z(t).$$

Редослед једначина у скупу једначина може да буде било какав, уосталом, као и у целом скупу за опис једног основног ентитета.

У скупу S4 се даје домен као и код описа криве у равни. Почетна вредност домена се обележава са додатком слова „m”, а крајња вредност са додатком слова „n” ознаци параметра (на пример  $a_m=c+2.0*d(12)$ ;  $a_n=a_m(15)+20.0$ ).

На исти начин може се формирати и израз код описа ентитета функцијама ( $2D/3D$ ), домена или кодомена, као и код дефинисања вредности за координате и друге параметре за опис у S4 ентитета дуж, круг и лук.

Иначе, може се код описа неког ентитета доделити вредност било којој од карактеристика којима се описује тај ентитет преузимањем исте карактеристике или друге (одговарајуће) карактеристике од неког другог дефинисаног ентитета. У свим изразима, где се преузима вредност неке од карактеристика (карактеристичне тачке, координате карактеристичне тачке, угла, дужине, пречника, полупречника, функције по којој се израчунавају координате – домен и кодомен на задатом домену, боја, тип линије, итд.), као и код описа помоћу релација (тангентно, управно, паралелно и слично), мора да се наведе и ентитет од

кога се та карактеристика преузима, односно ентитет са којим је ентитет који се описује у неком релационом односу (управно, паралелно и слично). Може да се и изостави навођење ентитета од кога се та карактеристика преузима ако се позива карактеристична координата или други карактеристични геометријски параметар од истог ентитета (који се описује).

Уместо директног додељивања идентификације ентитета из којег се позива нека карактеристика, може се њено дефинисање извршити и путем другачије селекције. На пример, може се дефинисати део равни (простора) у којем се налази бар једна тачка ентитета од којег се жели преузети нека (наведена) карактеристика.

Скуп  $S_4$  као део колекције  $S$  за опис једног од основних ентитета – дужи, лука, круга или друге криве линије јесте један низ елемената из скупа карактеристика и релационих исказа, али тако одабран да је довољан за потпуно дефинисање датог ентитета. Притом, карактеристике и релациони искази за дефинисање појединог ентитета, једначине за домен и опис кодомена могу да се нижу на разне начине. Постоји више могућности да се од елемената скупа, у коме су све карактеристике за дефинисање неког од ентитета, добије подскуп елемената за дефинисање ентитета који се тренутно описује. Притом је скуп карактеристика за дефинисање ентитета, из кога се узимају елементи за примарни (геометријски) опис различит за сваки основни ентитет.

Опис модификација (пресликавање, ротација итд.) може се такође дефинисати коришћењем елемената из колекције за опис основних ентитета. Овде се даје једно извођење описа модификација на примеру осносиметричног пресликавања код  $2D$  моделирања и цртања.

Командна линија за пресликавање једног ентитета може се дати на два начина. Може да са задаје број ентитета  $k$  који се пресликава у ентитет са редним бројем  $i+1$  са симетралом датом на исти начин као и код цртања дужи, и да се притом црта и та симетрала као и свака дуж са редним бројем  $i$ :

$i;MLM:k[\text{опис дужи}],$

где је  $MLM$  реч за пресликавање ентитета  $k$  у ентитет  $i+1$  са цртањем и симетрале  $i$ .

Друга могућност је да се не црта симетрала  $i$ . Само се црта пресликани ентитет  $k$  као нови ентитет под редним бројем  $i+1$  и тада се пресликавање дефинише на следећи начин:

$i;MM:k[\text{опис дужи}],$

где је  $MM$  реч за пресликавање ентитета  $k$  у ентитет  $i+1$  без цртања симетрале са редним бројем  $i$ .

Без обзира на то да ли се дата симетрала  $i$  црта или не црта у даљем дефинисању ентитета, могу се користити карактеристичне тачке, координате карактеристичних тачака и остали параметри ове симетрале као сваке друге усмерене дужи, као и релације положаја у односу на ову симетралу код дефинисања других ентитета.

Могуће је и скраћено писање, односно може се више командних линија ставити у једну на следећи начин:

$i+1, i+2, \dots, i+n; \text{MLM}: e_1, e_2, \dots, e_n$  [опис дужи]  
 или  
 $i+1, i+2, \dots, i+n; \text{MM}: e_1, e_2, \dots, e_n$  [опис дужи]  
 где су  
 $e_1, e_2, \dots, e_n$  – редни бројеви раније дефинисаних ентитета који се пресликавају,  
 $i+1, i+2, \dots, i+n$  – редни бројеви нових ентитета.

Ефекат ове заједничке командне линије је да ће се прсликати редом како су наведени и то: нови ентитет са редним бројем  $i+1$  ће бити прсликани ентитет  $e_1$ , нови ентитет са редним бројем  $i+2$  ће бити прсликани ентитет  $e_2$ , итд. Могуће је и још краће писање командне линије за пресликавање више ентитета:

$*; \text{MLM}: e_1, e_2, \dots, e_n$  [опис дужи]  
 или  
 $*; \text{MM}: e_1, e_2, \dots, e_n$  [опис дужи],

где реч  $*$  означава да су редни бројеви нових ентитета редом  $i+1, i+2$ , итд. до  $i+n$ , а  $n$  је укупан број ентитета наведених иза речи  $\text{MLM}$  и  $\text{MM}$ .

Ентитет са редним бројем  $i$  је дуж-симетрала која се стварно црта или је фиктивна ако се наведе реч  $\text{MM}$  за пресликавање.

Избор (селекција) ентитета који се пресликавају, осим навођења њихове идентификације, може се извести и на неки други начин.

Напомене наведене на крају тачке 1 важе и овде.

#### 4.1. Језик за $2D$ параметарско геометријско моделирање

Језик за  $2D$  параметарско геометријско моделирање приказан је као скраћена и програмирана варијанта.

##### Лексичка структура

Алфабет језика чине следећи скупови знакова

- велика и мала слова енглеског алфабета:  $A, B, \dots, Z$ ;  
 $a, b, \dots, z$ .
- знаменке:  $0, 1, 2, \dots, 9$
- остали знакови:  $\{ \} [ ] ( ) ; : , = \% \wedge + - * / < > \_ .$

Једнако значење има употреба великих и малих слова.

Речник чине следеће класе речи:

- константе (цели и реални бројеви и специјална константа  $\pi$ );
- кључне речи  
 $L, O, C, F, WL, WO, WC, WF, MM, MLM, M, N, C, XM, XN, XC, YM, YN,$   
 $YC, AM, AN, A, R, D, L$ ;
- стандардне функције:  $\sin, \cos, \text{tg}, \text{ctg}, \log, \ln, \text{arctg}$  итд.;
- идентификатори.



Имена промењивих могу да се састоје од слова и знаменки, при чему на првом месту мора да буде слово.

- остале речи:

Под осталим речима подразумевају се речи које не припадају ниједној од горе наведених класа, а то су:

- релације  $>$ ,  $<$  и једнакост  $=$ ,
- операције  $+$ ,  $-$ ,  $*$ ,  $/$  и степеновање  $\sim$ ,
- ознака за степене код углова  $\%$ ,
- специјалне функције

P, S, M, N, C, A, R, D, L, XM, XN, XC, YM, YN, YC, XP, YP, XS, YS, AM, AN

- релације положаја  $//$ ,  $IC$ ,  $I-$ ,
- тип линије  $\_$ ,  $\_$ ,  $\_$ , итд.
- маркери  $\{$ ,  $\}$ ,  $[$ ,  $]$ ,  $;$ ,  $:$ ,  $($ ,  $)$  и запета.

## Синтаксичка структура и семантика језика

Свака линија-команда описује један елементарни ентитет: дуж, кружницу, кружни лук или криву линију. У случају пресликавања групе ентитета могуће је скраћено писање, односно дефинисање више ентитета једном линијом. Делови линије-команде за опис ентитета могу се поделити у пет целина:

### 1. Идентификација

Овде је идентификациони број цели број у интервалу од 0 до 10000, а означава редни број ентитета.

### 2. Ознака ентитета

То је реч, односно симбол за врсту ентитета код цртања дужи, кружнице, кружног лука или друге криве линије. Код пресликавања то је реч којом се означава да се ради о пресликавању једног или групе ентитета.

### 3. Функције

Функције по којима се израчунавају координате  $x$ ,  $y$  код кривих линија. Код пресликавања то су редни бројеви једног или више ентитета који се пресликавају.

### 4. Опис ентитета

Опис ентитета довољним бројем карактеристичних тачака, параметара и релационих исказа, а код пресликавања опис симетрале као било које дужи. Код кривих линија овде се даје домен.

### 5. Тип линије ентитета

Уопште командна линија се може дати у следећем облику код цртања ентитета – дуж, кружница, кружни лук:

$\{i;LV:[Ti1=x_{i1},y_{i1}; \dots ;Ai1=ai1; \dots ;Ri1; \dots]LT\}$

где је:

$i$  – редни број ентитета

$LV$  – врста ентитета ( $LV$  је  $L$  – за дуж,  $O$  – за кружницу,  $C$  – за кржни лук, и  $F$  – за криву линију)

$T_{ij}$  – карактеристичне тачке за дефинисање ентитета

$A_{ij}$  – карактеристични параметри за дефинисање ентитета

$R_{ij}$  – релациони искази за дефинисање ентитета

$LT$  – тип линије

Могуће је да се координате карактеристичних тачака дају и појединачно:

$$\{i;LV:[X_{i1}=x_{i1};Y_{i1}=y_{i1}; \dots ;A_{i1}=a_{i1}; \dots ;R_{i1}; \dots ]LT\}$$

Код дефинисања криве линије дају се и функције по којима се израчунавају координате тачака за цртање криве линије:

$$\{i;LV:F1=f1;F2=f2[Gm=gm;Gn=gn]LT\}$$

где су  $F1$  и  $F2$  ознаке координата ( $X$  или  $Y$  на месту  $F1$ , односно  $F2$ ),

$f1$  и  $f2$ : аритметички изрази чије се израчунате вредности додељују одговарајућим, тј. наведеним координатама  $X$  и  $Y$  испред знака једнакости,

$Gm$  и  $Gn$ : границе ознака почетне и крајње вредности параметра који се даје код параметарског ефинисања кривих линија, или координата  $X$ , односно координата  $Y$  почетне и крајње тачке криве линије код непараметарског дефинисања кривих линија (уместо  $G$  даје се ознака параметра код параметарског дефинисања кривих линија, односно ознака  $X$  или  $Y$  код непараметарског дефинисања кривих линија),

$gm$  и  $gn$ : аритметички изрази чије се израчунате вредности додељују одговарајућем (наведеном) параметру односно наведеним координатама испред знака једнакости.

Аритметички изрази за израчунавање координата тачака и других карактеристичних параметара за дефинисање ентитета  $x_{ij}$ ,  $y_{ij}$ ,  $f1$ ,  $f2$ ,  $a_{i1}$ ,  $a_{i2}$ ,  $gm$ ,  $gn$  ... могу да буду састављени од:

- имена променљивих,
- константи,
- аритметичких операција  $+$ ,  $-$ ,  $*$ ,  $/$ ,  $\sim$ (степеновање),
- стандардних математичких функција  $\sin$ ,  $\cos$ ,  $\text{tg}$ ,  $\text{ctg}$ ,  $\arcsin$ , итд.
- $P_i$  константе.

Осим тога, могу се у аритметичким изразима користити и специјалне функције, као што су координате карактеристичних тачака:

$x_m(k)$  –  $x$ -координата почетне тачке дужи, лука или криве линије  $k$ ,

$y_m(k)$  –  $y$ -координата почетне тачке дужи, лука или криве линије  $k$ ,

$x_p(k)$  –  $x$ -координата крајње тачке дужи, лука или криве линије  $k$ ,  
 $y_p(k)$  –  $y$ -координата крајње тачке дужи, лука или криве линије  $k$ ,

$x_c(k)$  –  $x$ -координата центра лука или круга  $k$ ,  
 $y_c(k)$  –  $y$ -координата центра лука или круга  $k$ ,

$x_s(k)$  –  $x$ -координата средине лука или дужи  $k$ ,  
 $y_s(k)$  –  $y$ -координата средине лука или дужи  $k$ ,

$x_p(k,l)$  –  $x$ -координата пресека ентитета  $k$  ентитета  $l$ ,  
 $y_p(k,l)$  –  $y$ -координата пресека ентитета  $k$  ентитета  $l$ .

У аритметичким изразима могу се користити и остали карактеристични параметри:

$A(k)$  – угао оријентисане дужи  $k$  у односу на  $x$ -осу или централни угао лука  $MCN$ ,  
 $R(k)$  – полупречник лука или круга  $k$ ,  
 $D(k)$  – пречник лука или круга  $k$ ,  
 $L(k)$  – дужина дужи или тетиве лука  $k$ ,  
 $am(k)$  – угао дужи  $CM$  у односу на  $x$ -осу лука  $k$ ,  
 $an(k)$  – угао дужи  $CN$  у односу на  $x$ -осу лука  $k$ .

Код дефинисања тачака ентитета  $M$ ,  $N$  или  $C$  може да им се придружи и скуп координата на следећи начин  $T_{ij}=x_{ij},y_{ij}$ .

Ово значи: израчунај вредности аритметичких израза  $x_{ij}$  и  $y_{ij}$  и обе вредности као уређени пар придружи тачки за дефинисање ентитета  $T_{ij}$ .

Могуће су комбинације када се уместо посебног аритметичког израза за  $x$  и  $y$  координату даје вредност скупа координата (уређени пар) за неку карактеристичну тачку  $M$ ,  $N$ ,  $C$  итд. додељивањем вредности скупа координата тачака раније дефинисаних ентитета. Тада се уместо посебног аритметичког израза за сваку координату даје скуп координата неке карактеристичне тачке, додељивањем скупа координата тачака, других дефинисаних ентитета  $T_{ij}=t_i(k)$ , где вредност уређених парова координата  $t_i(k)$  може да буде:

$M(k)$  – почетна тачка дужи, лука или криве линије  $k$ ,  
 $N(k)$  – крајња тачка дужи, лука или криве линије  $k$ ,  
 $C(k)$  – тачка центра круга или лука  $k$ ,  
 $S(k)$  – средња тачка дужи или лука  $k$ ,  
 $P(k,l)$  – тачка пресека ентитета  $k$  и ентитета  $l$ .

Релације  $i$ -тог ентитета према раније дефинисаном ентитету могу да буду:

$//(k)$  – паралелно са ентитетом  $k$ ,  
 $I-(k)$  – управно на ентитет  $k$ ,  
 $IC(k)$  – тангентно на ентитет  $k$ .

Релациони искази ближе одређују скуп тачака ентитета који се дефинише, где се од два скупа тачака издваја један (жељени) скуп тачака. Притом, овде постоје два типа релационих исказа.

Један тип релационих исказа се пише уопште као  $g < h$  или  $g > h$ , где су  $g$  и  $h$  у општем случају аритметички изрази који могу да буду састављени на исти начин као што је наведено код координата и других параметара за дефинисање ентитета. У изразима  $g$  и (или)  $h$  се обавезно налази нека координата неке карактеристичне тачке и (или) други карактеристични параметар.

Други тип релационог исказа је  $g >$  или  $g <$ . Овде се поставља услов да је нека координата неке карактеристичне тачке већа или мања, односно неки параметар већи или мањи.

Командна линија за пресликавање једног ентитета може се дати на два начина. Може да се задаје број ентитета  $k$  који се пресликава у ентитет са редним бројем  $i+1$  са симетралом која се дефинише на исти начин као и свака усмерена дуж, а да се притом црта и та симетрала као и свака дуж са редним бројем  $i$ :

$$\{i;MLM:k[\text{дефиниција дужи}]LT\}$$

где је MLM реч за пресликавање ентитета  $k$  у ентитет  $i+1$  са цртањем и симетрале  $i$ .

Друга могућност је да се не црта симетрала  $i$ . Само се црта пресликани ентитет  $k$  као нови ентитет под редним бројем  $i+1$  и тада се пресликавање дефинише на следећи начин:

$$\{i;MM:k[\text{дефиниција дужи}]\}$$

где је MM реч за пресликавање ентитета  $k$  у ентитет  $i+1$  без цртања симетрале са редним бројем  $i$ .

Без обзира на то да ли се симетрала црта или не црта, у даљем дефинисању ентитета могу се користити карактеристичне координате и параметри ове симетрале као сваке друге усмерене дужи.

Могуће је и скраћено писање, односно може се више командних линија за пресликавање ставити у једну на следећи начин:

$$\{i+1,i+2, \dots ,i+n;MLM:e_1,e_2, \dots ,e_n[\text{дефиниција дужи}]LT\}$$

или

$$\{i+1,i+2, \dots ,i+n;MM:e_1,e_2, \dots ,e_n[\text{дефиниција дужи}]\}$$

где су

$e_1, e_2, \dots , e_n$  – редни бројеви раније дефинисаних ентитета који се пресликавају,

$i+1, i+2, \dots , i+n$  – редни бројеви нових ентитета.

Ефекат ове заједничке командне линије је да ће се пресликати редом како су наведени и то: нови ентитет са редним бројем  $i+1$  ће бити пресликани ентитет  $e_1$ , нови ентитет са редним бројем  $i+2$  ће бити пресликани ентитет  $e_2$ , итд.

Могуће је и још краће писање командне линије за пресликавање више ентитета:

{\*;MLM:e1,e2, ... ,en[дефиниција дужи]LT}

или

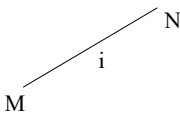
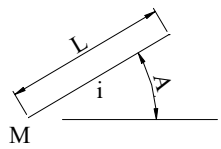
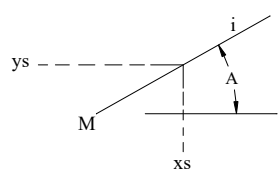
{\*;MM:e1,e2, ... ,en[дефиниција дужи]}

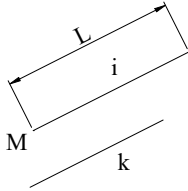
где реч \* означава да су редни бројеви нових ентитета редом  $i+1$ ,  $i+2$ , итд. до  $i+n$ , а  $n$  је укупан број ентитета који се пресликавају. Ентитет са редним бројем  $i$  је дуж-симетрала која се стварно црта или је фиктивна ако се наведе реч MM за пресликавање .

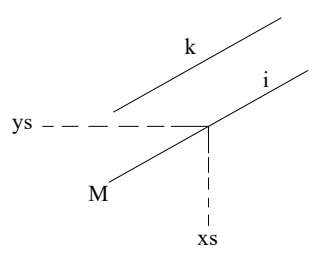
Основни ентитети дуж, кружни лук, кружница или крива линија могу да буду и фиктивни, тј. да се не цртају, али координате карактеристичних тачака као и саме тачке као скуп координата и други параметри тих фиктивних ентитета могу да се користе у опису других ентитета. Испред ознаке ентитета, ако се дефинише фиктивни ентитет, ставља се ознака W. Ови фиктивни ентитети су понекад потребни ради дефинисања других ентитета на једноставнији начин.

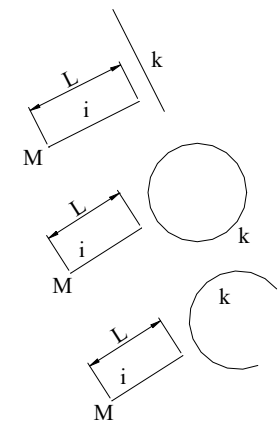
## 5. ДЕФИНИЦИЈЕ КОМАНДИ (КОМАНДНИХ ЛИНИЈА) ЗА МОДЕЛИРАЊЕ ЕНТИТЕТА

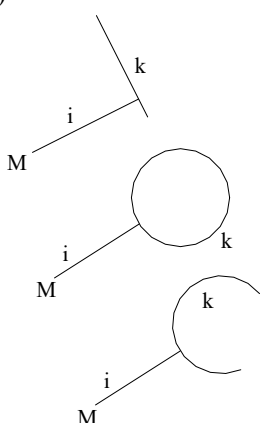
### 5.1. Оријентисана дуж

<p>1)</p> 	<p><b>Општа дефиниција команде (командне линије)</b>  <b>i;L: M=am1,am2;N=an1,an2 LT</b></p> <p><b>Примери командних линија за моделирање</b>  <u>дужи</u>            11:L: M=1,6;n=50,80            12:L: N=c+2,40;M=xm(2)+20,ym(2)            13:L: M=a*cos(bl),c!;N=d,ym(3)+20.0            14:L: M=(xs(1)+50)*sin(pi/2),c;N=P(2,3)            15:L: M=N(3);N=c(1)            16:L: M=M(2);N=a*tg(beta+gama),30            17:L: N=xm(1)+ym(2),ym(3)+20;M=xn(9),yn(9)+l</p>
<p>2)[56]</p> 	<p><b>Општа дефиниција команде (командне линије)</b>  <b>i;L: M=am1,am2;L=a!;A=aa LT</b></p> <p>у другој варијанти даје се командна линија за дуж специфицирањем крајње тачке N, дужине L и угла A:  <b>i;L: N=an1,an2;L=a!;A=aa LT</b></p> <p>у трећој варијанти даје се командна линија за дуж специфицирањем средње тачке S, дужине L и угла A:  <b>i;L: S=as1,as2;L=a!;A=aa LT</b></p> <p><b>Примери командних линија за моделирање</b>  <u>дужи</u>            18:L: M=2,2;L=20,5;A=pi/6            19:L: L=(xm(15)-xn(15))*sin(beta);A=PI;M=S(1)            20:L: M=xm(1)+10,ym(1);A=45%;L=a+b-6            21:L: M=N(4);L=L(5);a=30%            22:L: M=N(2);L=xm(4)-xm(3);A=A(1)+(pi/6)</p>
<p>3)</p> 	<p><b>Општа дефиниција команде (командне линије)</b>  <b>i;L: M=am1,am2;A=aa;xs=as1 LT</b>  <b>i;L: M=am1,am2;A=aa;ys=as2 LT</b></p> <p><b>Примери командних линија за моделирање</b>  <u>дужи</u>            11:L: M=20,30;A=pi/6;xs=25            12:L: M=a*cos(bl),c!;A=45%;xs=xm(12)+5            13:L: M=N(3);A=30%;ys=ym(13)+10            14:L: M=xm(1)+b,ym(3)+20;A=pi/6;ys=ym(14)+bl</p>

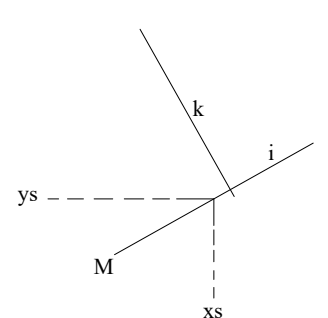
<p>4)[56,62,63]</p> 	<p><b>Општа дефиниција команде (командне линије)</b>  <b>i;L: M=am1,am2;L=al;//(k); релациони исказ (тип1, тип2 или тип3) LT</b></p> <p>У другој варијанти даје се командна линија за дуж и специфицирањем крајње тачке N, дужине L, релације паралелно, и релационог исказа за положај:  <b>i;L: N=an1,an2;L=al;//(k); релациони исказ (тип1, тип2 или тип3) LT</b></p> <p>У трећој варијанти даје се командна линија за дуж и специфицирањем средње тачке S, дужине L, релације паралелно, и релационог исказа за смер:  <b>i;L: S=as1,as2;L=al;//(k); релациони исказ (тип1, тип2 или тип3) LT</b></p> <p>Ентитет k, из опште дефиниције команде (командне линије) је друга дуж.</p> <p><b>Примери командних линија за моделирање дужи</b>  23;L: M=30,10;//(2);L=25;xp&gt;xm  24;L: M=(d1+d2)*2,d3;//(5);L=l(5); yp&lt;ym+yc(3)  25;L: //(10);L=bl;XM=xm(10)+l0;YM=ab;yp&lt;bl  26;L: M=N(5);//(15); L=20;xp&gt;  27;L: M=50,60;//(7);L=80;N M(7)</p> <p>Релациони исказ N M(7) значи да је крајња тачка дужи која се моделира ближа почетној тачки дужи 7.</p>
---	---

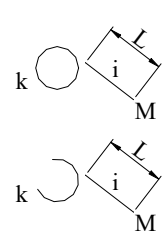
<p>5)</p> 	<p><b>Општа дефиниција команде (командне линије)</b>  <b>i;L: M=am1,am2;//(k);xs=as1 LT</b>  <b>i;L: M=am1,am2;//(k);ys=as2 LT</b></p> <p><b>Примери командних линија за моделирање дужи</b>  11;L: M=30,40;//(10);xs=35  12;L: M=a*cos(bl),c1;//(2);xs=xm(12)+5  13;L: M=N(3);//(3);ys=105  14;L: M=xm(1)+b,ym(3)+20;//(7);ys=ym(14)+(bl/2)</p>
---	--

<p>6)[56,63]</p> 	<p><b>Општа дефиниција команде (командне линије)</b>  <math>i;L: M=am1,am2;l-(k);L=al</math>; релациони исказ (тип1, тип2 или тип3) LT</p> <p>Друга варијанта је командна линија за дуж <math>i</math> специфицирањем крајње тачке <math>N</math>, релације управно, дужине <math>L</math> и релационог исказа за дефинисање положаја:  <math>i;L: N=an1,an2;l-(k);L=al</math>; релациони исказ (тип1, тип2 или тип3) LT</p> <p>Ентитет <math>k</math>, из опште дефиниције команде (командне линије) је друга дуж, кружница или кружни лук.</p> <p><b>Примери командних линија за моделирање дужи</b>  1;L: <math>M=3,3;l-(2);L=20;xp&lt;xm</math>  2;L: <math>N=c+2,d;l=L(2);l-(3);yp+2&gt;5</math>  3;L: <math>M=N(3);i-(4);L=bl+L(3);yp&gt;ym(3)+3,0</math>  4;L: <math>M=xm(l2),ym(l2)+2;l=30;l-(7);YN&gt;</math></p>
--	--

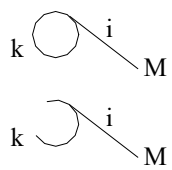
<p>6a)</p> 	<p><b>Општа дефиниција команде (командне линије)</b>  <math>i;L: M=am1,am2;l-(k)</math> LT  <math>i;L: N=an1,an2;l-(k)</math> LT</p> <p><b>Примери командних линија за моделирање дужи</b>  1;L: <math>M=3,3;l-(2)</math>  2;L: <math>N=c+2,d;l-l-(5)</math>  3;L: <math>M=N(3);l-(3)</math></p> <p>Дефиниције дужи дате овде су скраћене. дефиниције дужи датих под б); изостављен је параметар дужина и релациони исказ за дефинисање смера, подразумева се да друга тачка (почетна или крајња која није наведена у дефиницији) лежи на ентитету у односу на који је управна дуж која се дефинише; дужина је овде растојање од дефинисане тачке до ентитета у односу на који је управна.</p>
--	--

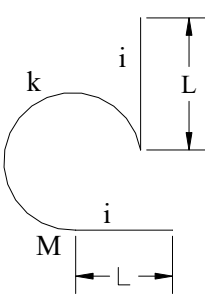


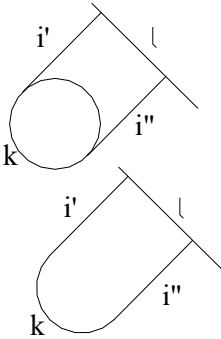
<p>7)</p> 	<p><b>Општа дефиниција команде (командне линије)</b>  <b>i;L: M=am1,am2;l-(k);xs=as1 LT</b>  <b>i;L: M=am1,am2;l-(k);ys=as2 LT</b></p> <p><b>Примери командних линија за моделирање</b>  <u>ДУЖИ</u>  11;L: M=35,45;l-(10);xs=50  12;L: M=a*sin(beta),c;l-(2);xs=xm(12)+bl  13;L: M=N(3);l-(3);ys=85  14;L: M=xm(1)+b,ym(3)+20;l-(7);ys=ym(14)+bl</p>
---	---

<p>8)[56,63]</p> 	<p><b>Општа дефиниција команде (командне линије)</b>  <b>i;L: M=am1,am2;lC(k);L=a;l; релациони исказ</b>  <b>(тип1, тип2 или тип3) LT</b></p> <p><b>Примери командних линија за моделирање</b>  <u>ДУЖИ</u>  11;L: M=l,6;l=l;lC(2);A&lt;arctg((yc(l)-ym)/(xc(l)-xm))  12;L: M=m(l);L=l*sin(ax);lC(4);A&lt;  13;L: M=a*cos(bl),e1+2.0;lC(3);L=e1+e2;A&gt;  14;L: M=xn(4)+20,yn(4);lC(7);L=80;NlC(5)</p> <p>Положај дужи у односу на лук или круг се специфицира релационим исказом да је угао дужи A већи или мањи од угла дужи M(i)-C(k) – пример дужи 11;  - ако се дефинише релациони исказ A&lt; (дуж 12), то је дуж са мањим углом;  - ако се специфицира релациони исказ A&gt; (дуж 13), то је дуж са већим углом.</p>
--	---

<p>8a)</p>	<p><b>Општа дефиниција команде (командне линије)</b>  <b>i;L: M=am1,am2;lC(k); релациони исказ</b>  <b>(тип1,тип2 или тип3) LT</b></p>
------------	--

	<p><b>Примери командних линија за моделирање</b></p> <p><u>ДУЖИ</u>  I4:L: M=xn(5),b-c;lC(4);A&gt;  I5:L: M=(xm(7)-xn(7))*sin(beta);A&lt;  I6:L: A&gt;lC(6);M=P(4,5)</p> <p>Ако се дужина не дефинише, подразумева се да је недефинисана тачка N на ентитету k; дефиниција дужи дате овде је скраћена; дефиниција дужи дате под 8); изостављен је параметар дужина; подразумева се да друга тачка (почетна или крајња која није наведена у дефиницији) лежи на ентитету у односу на који је тангентна дуж која се дефинише; дужина је овде растојање од дефинисане тачке до ентитета у односу на који је тангентна.</p>
---	---

<p>9)</p> 	<p><b>Општа дефиниција команде (командне линије)</b>  i;L: M=am1,am2;L=a;lC(k); релациони исказ (тип1,тип2 или тип3) LT</p> <p><b>Примери командних линија за моделирање</b></p> <p><u>ДУЖИ</u>  I1:L: M=N(9);L=20;lC(9);xn&gt;xm  I2:L: M=(d1+d2)*2,d3;lC(7);L=(5);ym&lt;yn  I3:L: lC(10);L=b;lXm=xm(10);Ym=ym(10);ym&lt;yn</p> <p>Ако се тачка M налази на ентитету k, тада мора да се дефинише смер релационим исказом.</p>
--	--

<p>10)</p> 	<p><b>Општа дефиниција команде (командне линије)</b>  i;L: lC(k);l-(l); два релациона исказа (тип1,тип2 или тип3) LT</p> <p><b>Примери командних линија за моделирање</b></p> <p><u>ДУЖИ</u>  I1:L: lC(5);l-(9);ys&lt;xm&gt;xp  I2:L: lC(7);l-(10);ys&lt;xm&lt;xp</p> <p>Уколико дуж није оријентисана, довољан је један релациони исказ.</p>
--	---

<p>11)[63]</p>	<p><b>Општа дефиниција команде (командне линије)</b>  i;L:M=am1,am2;xn=an1;L=a;l; релациони исказ (тип1, тип2 или тип3) LT  i;L:M=am1,am2;yn=an2;L=a;l; релациони исказ (тип1, тип2 или тип3) LT</p>
----------------	--

	<p><b>Примери командних линија за моделирање дужи</b></p> <p>13: L: M=20,25;xp=38;L=25;yp&gt;  14: L: M=xm(5)+b,ym(5);xp=xm+c;L=d;yp&lt;  15: L: M=xc(2),yc(2)+b;yp=35;xp&lt;xm</p> <p>Ако је разлика координата <math>x_m</math> и <math>x_p</math>, односно <math>y_m</math> и <math>y_p</math>, једнака дужини <math>L</math>, онда није потребно специфицирати релациони исказ за положај недефинисане координате <math>y_p</math> односно <math>x_p</math>.</p>
--	--

## 5.2. Кружница

<p>1)</p>	<p><b>Општа дефиниција команде (командне линије)</b>  <b>i;O: C=ac1,ac2;D=ad LT</b></p> <p><b>Примери командних линија за моделирање кружнице</b></p> <p>16: O: c=(xm(3)+xp(3))/2,ym(3)+h1;D=d5 _ _  17: O: D=d(4);c=xc(1)+t1,yc(1)-t1  18: O: C=xp(2,5)+2.0,yp(2,5);D=20.0</p>
<p>2)</p>	<p><b>Општа дефиниција команде (командне линије)</b>  <b>i;O: C=ac1,ac2;R=ar LT</b></p> <p><b>Примери командних линија за моделирање кружнице</b></p> <p>11: O: C=e1,e2;R=r5 _ _  12: O: R=20.0+r2;C=xm(1)+b1,ym(1) _ _  13: O: C=c(1);r=r5  14: O: C=xc(1)+e1,yc(1)+d*cos(beta);r=R(2)  15: O: R=D(5)-D(4);xc=xc(1);yc=yc(1)</p>
<p>3)[63]</p>	<p><b>Општа дефиниција команде (командне линије)</b>  <b>i;O: C=ac1,ac2;IC(k); релациони исказ (тип2) LT</b></p>

	<p><b>Примери командних линија за моделирање кружнице</b></p> <p>I1:L: C=25,30;IC(7);R&gt;  I2:L: C=45,50;IC(8);D&lt;  I3:L: C=xm(l)+b,70;R&lt;;IC(9)</p>
--	---

<p>4)[63]</p>	<p><b>Општа дефиниција команде (командне линије)</b>  i;O: IC(k);IC(l);R=ar; релациони исказ (тип1,тип2 или тип3) LT</p> <p><b>Примери командних линија за моделирање кружнице</b></p> <p>I1:L: IC(4);IC(7);R=l5;yc&gt;  I2:L: IC(5);IC(8);R=a;yc&lt;  I3:L: IC(9);IC(2);R=R(l1)+4;yc&lt;yc(9)</p>
---------------	--

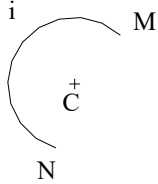
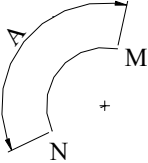
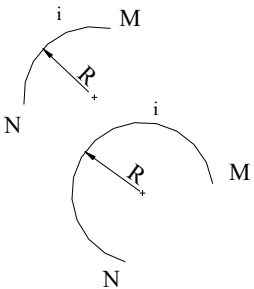
<p>5)[63]</p>	<p><b>Општа дефиниција команде (командне линије)</b>  i;O: IC(k);IC(l);D=ad; релациони исказ (тип1,тип2 или тип3) LT</p>
---------------	--

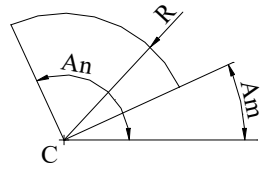
	<p><b>Примери командних линија за моделирање кружнице</b>  <math>11; L: IC(4); IC(7); D=20; yc &gt;</math>  <math>12; L: IC(5); IC(8); D=b; yc &lt;</math></p>
--	--

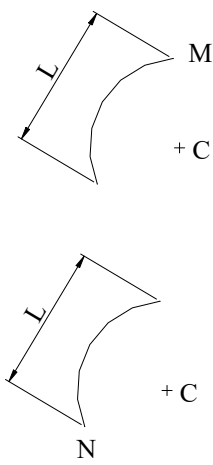
<p>6)[63]</p>	<p><b>Општа дефиниција команде (командне линије)</b>  <math>i; O: IC(k); IC(l);</math> релациони исказ (тип2) LT</p> <p><b>Примери командних линија за моделирање кружнице</b>  <math>14; O: IC(3); IC(5); R &gt;</math>  <math>15; O: IC(2); IC(4); D &lt;</math></p>
---------------	--

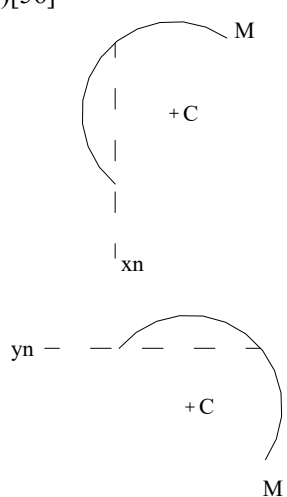
<p>7)</p>	<p><b>Општа дефиниција команде (командне линије)</b>  <math>i; O: IC(j); IC(k); xc=ac1</math> LT  <math>i; O: IC(j); IC(k); yc=ac2</math> LT</p> <p><b>Примери командних линија за моделирање кружнице</b>  <math>12; O: IC(4); IC(5); xc=20.0</math>  <math>13; O: IC(5); IC(6); yc=ym(7)+10.0</math></p>
-----------	--

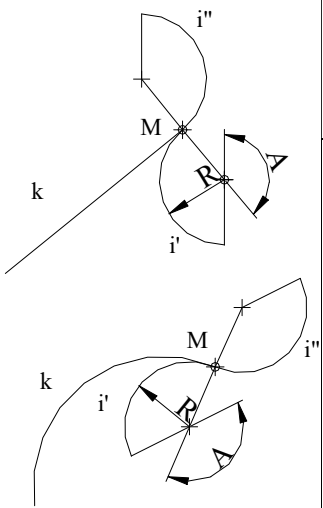
### 5.3. Кружни лук

<p>1)</p> 	<p><b>Општа дефиниција команде (командне линије)</b>  <b>i;C: M=am1,am2;N=an1,an2;C=ac1,ac2 LT</b></p> <p><b>Примери командних линија за моделирање кружног лука</b>            11;C: M=xm(5),ym(5)+b;C=c(3);N=xm(5)-b,ym(5)            12;C: M=xn(6)+b,yn(6);N=M(1);C=(xm+xn)/2,(ym+yn)/2            13;C: C=c(1);xn=xm(16)+2;yn=ym(16)+2;M=M(13)            14;c:m=a*cos(b),ym(11);C=xm+ra,ym-ra;N=xc-ra,yc</p>
<p>2)[62]</p> 	<p><b>Општа дефиниција команде (командне линије)</b>  <b>i;C: M=am1,am2;N=an1,an2;A=aa LT</b></p> <p><b>Примери командних линија за моделирање кружног лука</b>            15;c: M=20+b,c;N=n(5);A=45% __            16;C: A=a(5);N=xm(5),xn(5)+g;l;M=(a+d)/2.0,d*cos(gama)            17;C: M=5,5;n=6,6;A=am(5)+pi _ _            18;C: A=a(4)+(pi/4);M=xm(2)+30,ym(2);N=xm,ym-30 _</p> <p>Угао се даје у радијанима; ако се даје у степенима, онда мора да се нагласи додавањем на крају симбола %, као у примеру лука број 15.</p>
<p>3)[56,62]</p> 	<p><b>Општа дефиниција команде (командне линије)</b>  <b>i;C:M=am1,am2;N=an1,an2;R=ar; релациони исказ (тип1,тип2 или тип3) LT</b></p> <p><b>Примери командних линија за моделирање кружног лука</b>            19;C: M=40,40;n=20,20;r=30 __            20;C: M=a,b;N=a+c,b-c;N=s(3);R=d5/2;yc&gt;(ym+yn)/2            21;C: M=30,20;N=n(7);r=D4/2;L&lt;            22;C:M=50,70;N=90,120;r=r(4);C N(5)</p> <p>Ако је центар на средини дужи која спаја тачке М и N (<math>MN=2 \cdot R</math>), тада је сувишно дати релацију за положај центра лука.</p>

<p>4)</p> 	<p><b>Општа дефиниција команде (командне линије)</b>  <b>i;C:C=ac1,ac2;AM=aam;AN=aan;R=ar LT</b></p> <p><b>Примери командних линија за моделирање</b>  <b>кружног лука</b>  18;C: C=20,20;am=30%;an=60%;R=10  19;C: C=s(5);am=0;an=pi;R=l(5)/2  20;C: am=pi/2;an=pi;r=d4/2;C=C(6) __</p>
---	--

<p>5)</p> 	<p><b>Општа дефиниција команде (командне линије)</b>  <b>i;C: C=ac1,ac2;M=am1,am2;L=al LT</b>  <b>i;C: C=ac1,ac2;N=an1,an2;L=al LT</b></p> <p><b>Примери командних линија за моделирање</b>  <b>кружног лука</b>  15;C: M=30,30;c=20,20;L=1.5  16;c: M=20+b,c;C=S(5);L=D(4)-20  17;C:L=L(5)+L(4);M=xm(5),xn(5)+g;C=xc(6),yc(6)  18;C:n=b,e;L=L(5)+R(4);C=xc(2)+el,yc(2)-el</p>
--	--

<p>6)[56]</p> 	<p><b>Општа дефиниција команде (командне линије)</b>  <b>i;C:C=ac1,ac2;M=am1,am2;xn=an1; релациони исказ (тип1,тип2 или тип3) LT</b>  <b>i;C:C=ac1,ac2;M=am1,am2;yn=an2; релациони исказ (тип1,тип2 или тип3) LT</b></p> <p><b>Примери командних линија за моделирање</b>  <b>кружног лука</b>  15;C: M=30,20;c=20,20;xn=15;L&gt;  16;c: M=a,b+8;C=a,b;xn=a-4;L&lt;  17;C:M=xm(5),xn(5);C=xm(5),ym(5)+b;yn=yc+b/2;xn&lt;xc</p>
---	--

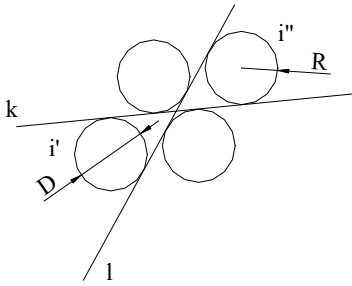
<p>7)[60,63]</p> 	<p><b>Општа дефиниција команде (командне линије)</b>  <math>i;C:M=am1,am2;IC(k);R=ar;A=aa</math>; релациони исказ (тип1,тип2 или тип3) LT  <math>i;C:N=an1,an2;T=N(i);IC(k);R=ar;A=aa</math>; релациони исказ (тип1,тип2 или тип3) LT</p>
	<p><b>Примери командних линија за моделирање кружног лука</b>  11;C: M=s(1);r=l;a=pi/3;IC(1);yp&lt;ym  12;C: M=s(1);r=l;a=pi/3;IC(1);yp&gt;ym  13;C: N=s(1);T=s(1);r=l;a=pi/3;IC(1);yp&lt;ym  14;C: N=s(1);T=s(1);r=l;a=pi/3;IC(1);yp&gt;ym  15;C: M=M(3);a=100%;r=r(5);iC(3);Yc&gt;(ym+yn)/2  16;C: M=xc(2)+r(2).yc(2);iC(2);r=0.5;A=120%;xc&gt;xm</p> <p>Ако се даје да је тачка М лука који се дефинише истовремено тангентна тачка, тада није потребно наглашавати која је тангентна тачка – подразумева се да је то тачка М.  Ако се даје тачка N лука који се дефинише, мора се дефинисати тангентна тачка као код примера 13 и 14.  Релационим исказом се дефинише положај где ће се лук цртати.</p>

## 5.4. Специфични примери дефинисања основних ентитета

- Примери дефиниција са релационим исказима где се користе релације *највеће* или *најмање*, као и релације *најдаље* и *најближе*

<p>7 кружница [40]</p>	<p><b>Општа дефиниција команде (командне линије)</b>  <math>i;O: IC(k);IC(l);D=ad</math>; релациони исказ (подтип тип2 или подтип тип3) LT  <math>i;O: IC(k);IC(l);R=ar</math>; релациони исказ (подтип тип2 или подтип тип 3) LT</p>
------------------------	---



	<p><b><u>Примери командних линија за моделирање кружнице</u></b></p> <p>11;0: IC(2);IC(4);R=15;yc&gt;&gt;  12;0: IC(3);IC(5);D=d1;yc&lt;&lt;  13;0: IC(7);IC(1);R=R(2)+2;xc&lt;&lt;  24;0: IC(10);IC(6);R=d1/2;C  M(10)  25;0: IC(12);IC(15);D=20;C__N(15)</p> <p>Објашњење:  Хс&gt;&gt; – ово значи хс највеће;  Ус&lt;&lt; – ово значи ус најмање;  -ако се дефинише релациони исказ ус&lt;&lt; издваја се круг i' на шеми;  -ако се дефинише релациони исказ ус&gt;&gt; издваја се круг i''.</p> <p>У примеру круга 24 дат је релациони исказ C  M(10), што значи да је тачка центра овог круга најближа почетној тачки ентитета (дужи)10.</p> <p>У примеру круга 25 дат је релациони исказ C__N(15), што значи да је тачка центра овог круга најдаља од крајње тачке ентитета (дужи) 15.</p>
---	--

- Примери дефиниција са релационим исказима где се користе сви типови и подтипови релационих исказа

<p>9 дуж [40]</p>	<p><b><u>Општа дефиниција команде (командне линије)</u></b>  i;l: M=am1,am2;IC(k);L=a;l; релациони искази (тип1, тип2, тип3 – два од ова три исказа) или (подтип тип2) или (подтип тип3) LT</p>
-------------------	---

	<p><b>Примери командних линија за моделирање дужи</b></p> <p>  ;L: M=20,30;L=70; C(4);N C(4);yp&gt;  2;L: M=N(3); C(2);L= l-50;N_C(2);yp&lt;</p> <p>Објашњење:</p> <ul style="list-style-type: none"> <li>  – ознака за ближе;</li> <li>_ – ознака за даље;</li> </ul> <p>-ако се дефинише релациони исказ N C(k) (N је ближе C(k)), издвајају се две дужи са положајем ближе кружности или кружном луку k и смером ка кружности или кружном луку;</p> <p>-ако се дефинише релациони исказ N_C(k) (N је даље од C(k)), издвајају се две дужи са положајем даље од кружности или кружног лука k усмерене од кружности или кружног лука.</p> <p>Још једним релационим исказом издваја се једна дуж:</p> <ul style="list-style-type: none"> <li>- овде се једним релационим исказом подтип типа 2 издваја једна дуж, на пример yp&gt;&gt;;</li> <li>- овде се једним релационим исказом подтип типа 3 издваја једна дуж, на пример N   (20,40).</li> </ul>
--	---

- Примери дефиниција са релационим исказима где се користе релације *најближе* или *најдаље*

<p>8 кружница</p>	<p><b>Општа дефиниција команде (командне линије)</b></p> <p>i;O:  C(k); C(l);R=ar; релациони исказ (подтип тип3) LT i;O:  C(k); C(l);D=ad; релациони исказ (подтип тип3) LT</p> <p><b>Примери командних линија за моделирање кружнице</b></p> <p>  ;C:  C(2); C(3);R=2;C_N(3)  2;C:  C(5); C(7);R=4;C  M(7)</p> <p>Објашњење:</p> <ul style="list-style-type: none"> <li>   – ознака за најближе;</li> <li>___ – ознака за најдаље;</li> </ul> <p>- ако се дефинише релациони исказ C  N(k) (C је најближе N(k)), издваја се круг i';</p> <p>- ако се дефинише релациони исказ C__N(k) (C је најдаље од N(k)), издваја се круг i''.</p>
-------------------	---

Даље се даје још једна дефиниција дужи без шеме.

**Општа дефиниција команде (командне линије)**

i;l: M=am1,am2;IC(k);xp=an1; релациони исказ тип1, тип2 или тип3 LT

i;l: M=am1,am2;IC(k);yp=an2; релациони исказ тип1, тип2 или тип3 LT

Може се дати командна линија за дуж и специфицирањем крајње тачке N, релације тангентно, координате xm односно ym, и положаја дужи релационим исказом:

i;l: N=an1,an2;IC(k);xm=am1; релациони исказ тип1 или тип2 или тип3 LT

i;l: N=an1,an2;IC(k);ym=am2; релациони исказ тип1 или тип2 или тип3 LT

Ентитет k, из опште дефиниције команде (командне линије) је друга кружница или кружни лук.

**Примери командних линија за моделирање дужи**

l1:l: M=l,b;IC(5);xp=3.0;A<

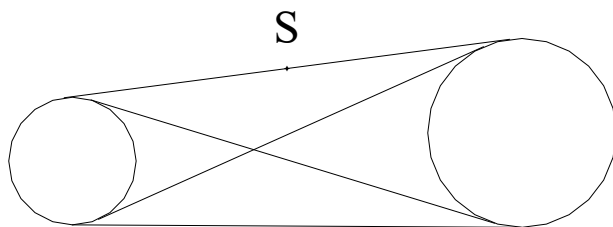
l2:l: M=xm(2)+20,ym(2);IC(6);yp=2;A>

l3:l: M=a\*cos(b1),c1;xp=xm(3)+20.0;IC(4);yp>

l4:l: yp=xm(l1)+b1;M=m(7);IC(3);A>

**5.5. Ентитети који су погодни за интерактивно моделирање са једним подтипом релационих исказа**

- Дуж тангентна на две кружнице или два кружна лука

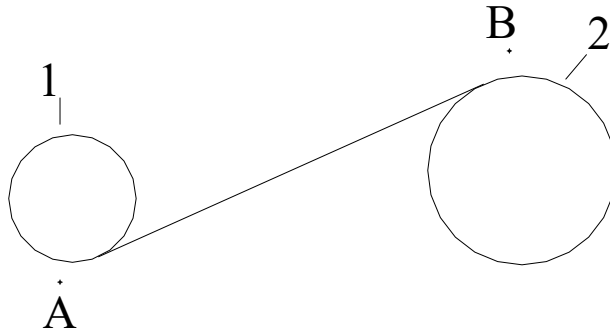


Слика 8. Дуж тангентно на две кружнице I

За спољне дужи лако је моделирање (горња шема):

IC(k);IC(l);S.||. тачки задатој мишем; xp>xm (|| – ознака за најближе; k и l су кружнице или кружни лукови које тангира дуж која се моделира).

Ова дефиниција може да се користи и за унутрашње дужи ако нису кружнице или кружни лукови које тангира дуж истог пречника (xp>xm је оријентација дужи).



Слика 9. Дуж тангентно на две кружнице II

Постоји решење у интерактивном раду (горња шема):

$IC(1);IC(2);M.||.A;N.||.B$

Ова дефиниција је слична оној датој у *AutoCAD*-у ; – овом дефиницијом се решава и оријентација.

Довољно је прецизно

$IC(1);IC(2);M.||.A;N.||.B$

Овде је довољно  $N.||.B$  зато што имају само два решења за тачку  $N$  на основу претходног описа дужи (овде би у софтверу морало да се предвиди да се прво решава релациони исказ  $M.||.A$ )

или

$IC(1);(C(2);M.||.A;un>$  .

Овде је довољно  $un>$  зато што имају само два решења за тачку  $N$  на основу претходног описа дужи (овде би у софтверу морало да се предвиди да се прво решава релациони исказ  $M.||.A$ )

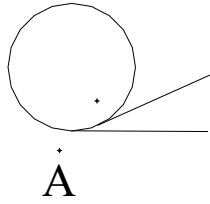
или

$IC(1);(C(2);M.||.A;un>uc(2)$ .

Овде је довољно  $un>uc(2)$  зато што имају само два решења за тачку  $N$  на основу претходног описа дужи (овде би у софтверу морало да се предвиди да се прво решава релациони исказ  $M.||.A$ ).

У софтверу морају да се посматрају сва три ентитета јер овде имају две блиске тачке

(морало би то да се зумира као што је дато на доњој шеми).



Слика 10: Дуж тангентно на две кружности III

## 5.6. Дефиниције за моделирање дужи паралелних $x$ или $y$ оси

Када се моделирају дужи које су паралелне  $x$  или  $y$  оси, могу се дати скраћене дефиниције[62]:

а)

**i; YL: M=am1,am2; yn=an2**

Овде је дуж означена са YL паралелна  $y$ -оси. Специфицира се почетна тачка и  $y$  координата крајње тачке.

б)

**j: XL: M=am1;am2;xn=an1**

Овде је дуж означена са XL паралелна  $x$ -оси. Специфицира се почетна тачка и  $x$  координата крајње тачке.

	<p>Када се дужи настављају као на шеми лево, две дефиниције за дужи (i) и (j) су[62]:</p> <p><b>i; YL: M=am1,am2; yn=an2</b></p> <p><b>j: XL: M=N(i);xn=an2</b></p> <p>Када се дужи настављају као на шеми или слично томе, дефиниције могу да буду још краће:</p> <p><b>i: YL: _;yn=an2</b></p> <p><b>j: XL: _;xn=an1</b></p> <p>Знак <b>_</b> означава да се специфицирана дуж наставља на претходну дуж.</p>
--	---

Када се задаје дужина уместо одговарајуће координате крајње тачке, тада су скраћене дефиниције:

а) скраћена дефиниција дуж  $y$ -осе

**i: YL: M=am1,am2; L=al; релациони исказ тип1, тип2 или тип3**

Овде се може специфицирати једноставан релациони исказ  $yn >$  (тип2) или  $yn > um$  (тип1). Очигледно да у релационом исказу мора да стоји она координата која је и  $y$  ознака дужи, тј. координата са ознаком осе у правцу које се моделира дуж.

б) скраћена дефиниција дуж  $x$ -осе

**j: XL: M=am1,am2; L=al; релациони исказ тип1, тип2 или тип3**

Још краће дефиниције су када се изостави и ознака и почетна тачка:

**\_;L=al;yn >**

односно са релационим исказом тип1

**\_;L=al;yn > um**

Овде се дуж наставља на претходну дуж. Зна се да је ово дуж YL по томе што у релационом исказу стоји ознака у.

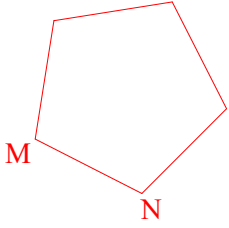
Слична је и скраћена дефиниција дуж x-осе.

`_;L=al;xn>`

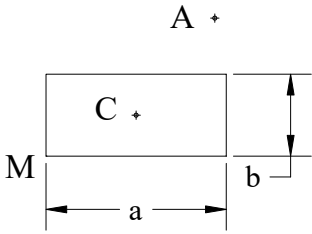
односно са релационим исказом тип1

`_;L=al;xn>xm .`

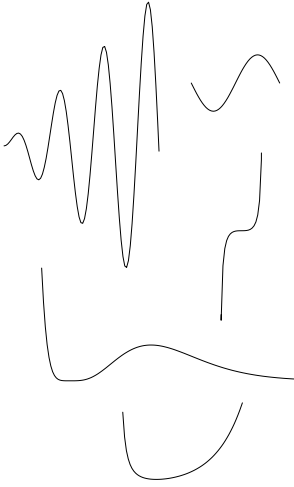

## 5.7. Многоугао

	<p><b>Општа дефиниција команде (командне линије)</b>  <i>i</i>;P: M=am1,am2;N=an1,an2 ;k=ak; релациони исказ тип1 или тип2 или тип3 LT</p>
	<p><b>Примери командних линија за моделирање</b>  <b>МНОГОУГЛА</b>          f1;P: M=1,6;n=50,80 ;k=5 ;yc&gt;ym          f2;P: N=c+2,40;M=xm(2)+20,ym(2);k=6 ;yc&gt;</p> <p>R – ознака за многоугао          k – број страница многоугла</p>

## 5.8. Правоугаоник

	<p><b>Општа дефиниција команде (командне линије)</b>  <i>i</i>; R: M=am1,am2;a=aa;b=ab; релациони искази (тип1, тип2, тип3 –два од ова три исказа) или (подтип тип2) или (подтип тип3) LT</p>
	<p><b>Примери командних линија за моделирање</b>  <b>Правоугаоника</b>          f1;R: M=20,20; a=30;b=10; C  M(5)          f2;R: M=40,30; a=f-e;b=f-c; C_N(7)          f3;R:M=20,20;a=10;b=20;xc&gt;;yc&gt;</p> <p>R – ознака за правоугаоник</p>

## 5.9. Крива линија

<p>a)</p> 	<p><b>Општа дефиниција команде (командне линије)</b>  <math>i;F: X=x;Y=f(x) X_m=ax_m;X_n=ax_n LT</math>  <math>i;F: Y=y;X=f(y) Y_m=ay_m;Y_n=ay_n LT</math></p> <p><b>Примери дефинисања криве линије</b>            11;F:x=x;y=x*(x-2.0) x<sub>m</sub>=20.0;x<sub>n</sub>=40.0            12;F:x=x;y=(sin(x+pi))*2.0 x<sub>m</sub>=0;x<sub>n</sub>=pi/2            13;F:y=a*tg(x*pi);x=x X<sub>n</sub>=a+c;x<sub>m</sub>=b*2            14;F:y=y;x=y~3-y~2-2 Y<sub>m</sub>=1.0;y<sub>n</sub>=6.0            15;F:Y=y;X=a*y~2-b*y-c y<sub>m</sub>=a-b;y<sub>n</sub>=a            16;F:X=x;y=r1*x+r2*y~2 x<sub>m</sub>=4.0;x<sub>n</sub>=120.0            17;F:x=x;y=(x-3)~4/2~x x<sub>m</sub>=a;x<sub>n</sub>=b</p> <p>Ово је дефинисање криве у непараметарском облику.            Код непараметарског дефинисања кривих за сваки <math>x</math> из дефинисаног интервала за <math>x</math> може да постоји само једна вредност <math>y</math>, односно за сваки <math>y</math> из дефинисаног интервала може да постоји само једна вредност <math>x</math>; аритметички израз за израчунавање једног од елемената <math>x</math> или <math>y</math> у уређеном пару мора да буде такав да искључује добијање више решења за <math>x</math> односно <math>y</math>.</p>
<p>б)</p> 	<p><b>Општа дефиниција команде (командне линије)</b>  <math>i;F:x=f1(t),y=f2(t) t_m=at_m;t_n=at_n LT</math></p> <p><b>Примери дефинисања криве линије</b>            18;F:y=a*sin(t);x=a*cos(t) t<sub>m</sub>=0.0;t<sub>n</sub>=pi            19;F:x=b+r*cos(a);y=c+r*sin(a) a<sub>m</sub>=pi;a<sub>n</sub>=2*pi            20;F:y=q*5;x=q~2 q<sub>m</sub>=a-c;q<sub>n</sub>=a            21;f:x=b*t;y=t~2 t<sub>m</sub>=1;t<sub>n</sub>= 2-1             22;F:x=sin(tt);y=tt t<sub>m</sub>=2*pi;t<sub>n</sub>=0            23;F:y=2~q;x=(5*sin(q))~2 q<sub>m</sub>=a;q<sub>n</sub>=b            24;f:x=2~t/t;y=t~2 t<sub>m</sub>=a;t<sub>n</sub>=b</p> <p>Ово је дефинисање криве у параметарском облику.            Код параметарског дефинисања кривих могу се добити и криве, где за једну вредност <math>x</math> или <math>y</math> у координате постоји више тачака са различитим припадајућим другим координатама; може да постоји једна иста координата у више уређених парова <math>(x,y)</math>, тј. тачака криве линије.</p>

## 6. ПРИМЕНА ЈЕЗИКА ЗА 2D ПАРАМЕТАРСКО ГЕОМЕТРИЈСКО МОДЕЛИРАЊЕ

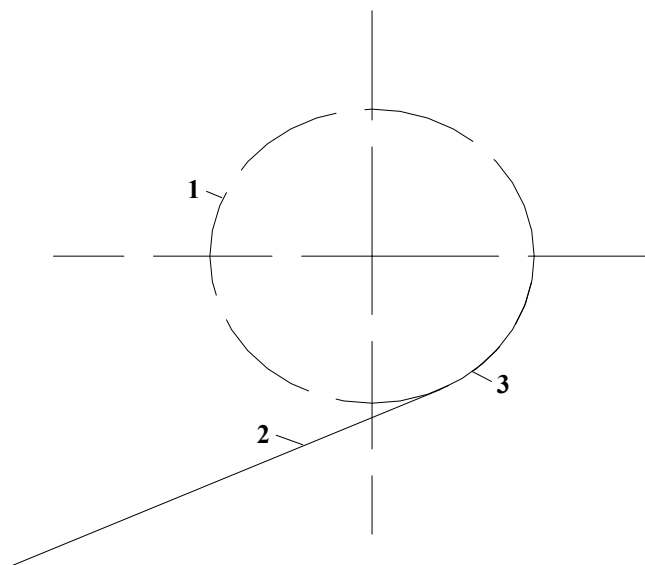
### 6.1. Примери описаних форми машинских елемената

	<p>Форма</p>
<p>Програм: чаура - ПОГЛЕД ОДОЗДО Б</p> <pre> {1:l:[m=150,150;n=xm(1),ym(1)]} {2:l:[m=xm(1)+k+2,ym(1);l=k+(d4/2)+5;a=pi]_} {3:l:[m=xm(1)+k,ym(1);a=pi/2;l=((d4/2)-2-k-2)~0.5]} {4:c:[m=n(3);n=xm(3)-k-(d4/2),ym(3);r=d4/2]} {5:c:[c=c(4);n=xn(4)+2,yn(4);a=a(4)]} {6:l:[m=m(5);n=xm(3)-2,ym(3)]} {7:o:[c=xm(1)+(d5/2)*cos(pi/3),ym(1)+(d5/2)*sin(pi/3);d=d6]} {8:c:[n=xm(1)-(d5/2),ym(1);c=c(4);a=pi-(pi/7)]_} {9:o:[c=c(7);r=(d(5)/2)-(d5/2)]} {10:o:[c=xc(4)-(d5/2.0),yc(4);r=r(7)]} {11:o:[c=xc(4)-(d5/2.0),yc(4);r=r(9)]} {12:o:[c=c(4);d=d1]} {13:o:[c=c(4);d=d1+4]} {14:l:[m=xc(4),yc(4)-(d4/2.0)-2.0;l-(2);l=d4+4;yn&gt;ym]_} {15:l:[m=c(7);i-(12);l=r(9)+2;xn&lt;xm]} {16:l:[m=c(7);i-(12);l=r(9)+2;xn&gt;xm]} {*;MM:3:4:5:6:7:8:9:15:16[m=m(2);n=n(2)]} </pre>	



## 6.2. Фиктивни ентитети

Пример употребе фиктивних ентитета



Слика 11: Пример моделирања кружног лука помоћу фиктивне кружнице

1:W0: C=50,40;R=10

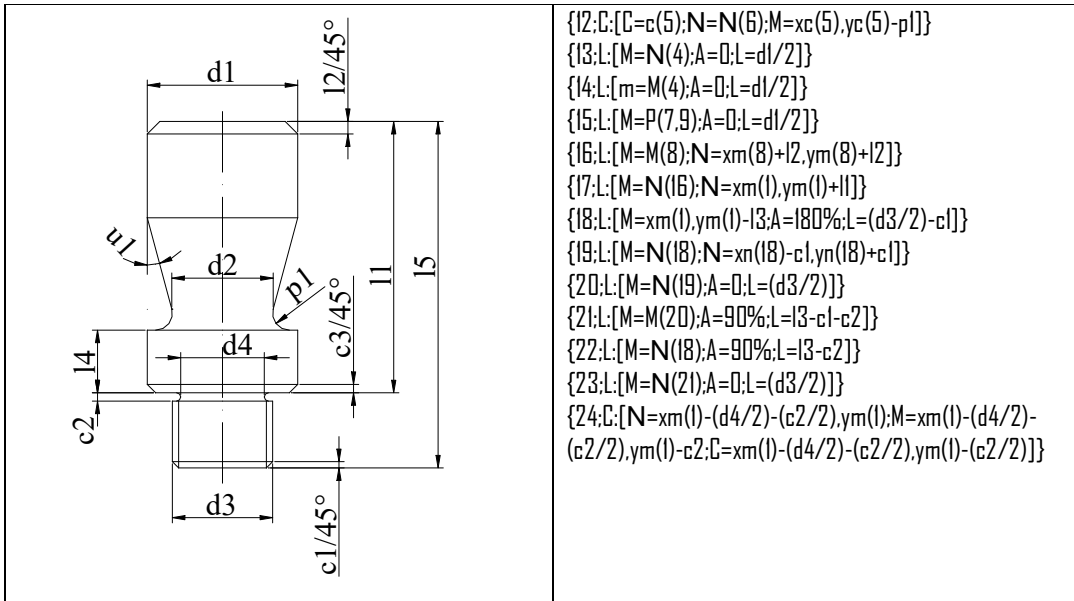
2:L: M=5,20;I0(I)

3:E: M=N(2);C=C(I);N=xс(I)+R(I),yc(I)

Овде (слика 11) фиктивна кружница 1 служи да се код дефинисања дужи 2 (тангентна на кружницу 1) добије почетна тачка N(2) кружног лука 3.

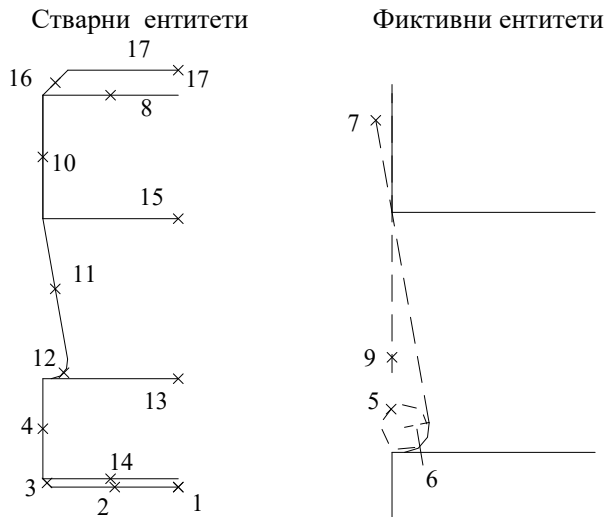
Примена фиктивних ентитета на једном фабричком типском елементу (чеп за алат)

Форма	Програм:
	{1:L:[M=100,100;N=xm(I),ym(I)]}
	{2:L:[M=m(I);a=180%;L=(d1/2)-c3]}
	{3:L:[M=N(2);N=xm(3)-c3,ym(3)+c3]}
	{4:L:[M=N(3);A=90%;L=l4-c3]}
	{5:W0:[C=xn(4)+((d1/2)-(d2/2))-pl,yn(4)+pl]}
	{6:WL:[M=C(5);L=pl;A=(ul)%]}
	{7:WL:[M=N(6);A=(90+ul)%;L=l1]}
	{8:L:[M=xm(I)-(d1/2),ym(I)+l1-l2;A=0;L=(d1/2)]}
	{9:WL:[M=m(8);/(4);yn<ym;L=l1]}
	{10:L:[M=m(8);N=P(7,9)]}
	{11:L:[M=P(7,9);N=M(7)]}



Да би се дефинисали ентитети 10, 11 и 12, овде су претходно дефинисани фиктивни ентитети 5, 6, 7 и 9 (на десном делу доње слике су обележени фиктивни ентитети).

Коришћењем фиктивних ентитета знатно се олакшава дефинисање ентитета 10, 11 и 12.



Слика 12: Коришћење фиктивних ентитета код моделирања машинског елемента чеп за алат

Да би било јасније, за овај пример ће се речима описати поступак моделирања, односно програм дат у претходној табели.

Ентитет 1 је дуж где су почетна и крајња тачка једнаке.

Ентитет 2 је дуж где је почетна тачка у почетној тачки ентитета 1; угао у односу на  $x$ -осу (позитиван смер) је 180 степени; дужина ове дужи је половина пречника, тј. димензија  $d1$  умањена за димензију  $c3$ .

Ентитет 3 је дуж где је почетна тачка једнака почетној тачки ентитета 2; крајња тачка је дефинисана  $x$  координатом која се добија када се од  $x$  координате почетне тачке ове дужи одузме димензија  $s_3$ , а у координата крајње тачке ове дужи је једнака у координати почетне тачке ове дужи увећана за димензију  $s_3$ .

Ентитет 4 је дуж чија је почетна тачка једнака крајњој тачки ентитета 3; угао у односу на  $x$ -осу је  $90$  степени; а дужина се добија израчунавањем разлике дужине  $l_4$  и димензије  $s_3$ .

Ентитет 5 је фиктивна кружница чији је центар такав да јој је у координата центра једнака  $x$  координати ентитета 4 увећана за разлику  $((d_1/2)-(d_2/2))$  и умањена за димензију  $p_1$ ; у координата тачке центра кружнице је у координата крајње тачке ентитета 4 увећана за димензију  $p_1$ .

Ентитет 6 је фиктивна дуж чија је почетна тачка једнака тачки центра ентитета 5 (кружнице); дужина дужи је једнака димензији  $p_1$ , а угао је дат у степенима и једнак је димензији  $u_1$ .

Ентитет 7 је фиктивна дуж чија је почетна тачка једнака крајњој тачки ентитета 6; угао је једнак изразу  $90+u_1$  и дат је у степенима; дужина је једнака димензији  $l_1$ .

Ентитет 8 је дуж којој је  $x$  координата почетне тачке једнака  $x$  координати почетне тачке ентитета 1, од које се одузима половина димензије  $d_1$ , а у координата почетне тачке је једнака у координати почетне тачке ентитета 1, којем је додата дужина  $l_1$  и одузета дужина  $l_2$ ; угао је једнак  $0$  и дат је у радијанима; дужина дужи је једнака половини димензије  $d_2$ .

Ентитет 9 је фиктивна дуж чија је почетна тачка једнака почетној тачки ентитета 8; паралелна је са ентитетом 4; у координата крајње тачке је мања од у координате почетне тачке; дужина дужи је једнака димензији  $l_1$ .

Ентитет 10 је дуж чија је почетна тачка једнака почетној тачки ентитета 8, а крајња тачка једнака пресечној тачки ентитета 7 и 8.

Ентитет 11 је дуж чија је почетна тачка једнака пресечној тачки ентитета 7 и 9, а крајња тачка једнака почетној тачки ентитета 7.

Ентитет 12 је кружни лук чија је тачка центра једнака тачки центра ентитета 5; крајња тачка је једнака крајњој тачки ентитета 6;  $x$  координата почетне тачке је једнака  $x$  координати центра ентитета 5, а у координата почетне тачке је једнака у координати центра ентитета 5 умањена за димензију  $p_1$ .

Ентитет 13 је дуж чија почетна тачка је једнака крајњој тачки ентитета 4; угао дужи у односу на  $x$ -осу је  $0$  радијана; дужина ове дужи је половина димензије  $d_1$ .

Ентитет 14 је дуж чија је почетна тачка једнака почетној тачки ентитета 4; угао у односу на  $x$ -осу је  $0$  радијана; дужина дужи је половина димензије  $d_1$ .

Ентитет 15 је дуж чија је почетна тачка једнака пресечној тачки ентитета 7 и 9; угао у односу на  $x$ -осу (позитиван смер) је  $0$  радијана и дужина дужи је половина димензије  $d_1$ .

Ентитет 16 је дуж чија је почетна тачка једнака почетној тачки ентитета 8;  $x$  координата крајње тачке је једнака  $x$  координати почетне тачке ентитета 8 увећаној за димензију  $l_2$ , а у координата крајње тачке је једнака у координати почетне тачке ентитета 8 увећаној за димензију  $l_2$ .

Ентитет 17 је дуж чија је почетна тачка једнака крајњој тачки ентитета 16;  $x$  координата крајње тачке је једнака  $x$  координати почетне тачке ентитета 1, а у координата крајње тачке је једнака у координати почетне тачке ентитета 1 увећаној за димензију  $l_1$ .

Ентитет 18 је дуж чија је  $x$  координата почетне тачке једнака  $x$  координати почетне тачке ентитета 1, а у координата почетне тачке је једнака у координати почетне тачке ентитета 1 умањеној за  $l_3$ ; угао дужи у односу на  $x$ -осу је  $180$

степенни; дужина дужи је једнака половини димензије  $d_3$  умањеној за димензију  $c_1$ .

Ентитет 19 је дуж чија је почетна тачка једнака крајњој тачки ентитета 18;  $x$  координата крајње тачке је једнака  $x$  координати крајње тачке ентитета 18 умањена за димензију  $c_1$ , а  $y$  координата крајње тачке је једнака  $y$  координати крајње тачке ентитета 18 увећана за димензију  $c_1$ .

Ентитет 20 је дуж чија је почетна тачка једнака крајњој тачки ентитета 18; угао у односу на  $x$ -осу је 0 радијана; дужина дужи је једнака половини димензије  $d_3$ .

Ентитет 21 је дуж чија је почетна тачка једнака почетној тачки ентитета 20; угао је 90 степени; дужина дужи је једнака димензији  $l_3$  умањеној за димензије  $c_1$  и  $c_2$ .

Ентитет 22 је дуж чија је почетна тачка једнака крајњој тачки ентитета 18; угао у односу на  $x$ -осу је 90 степени; дужина дужи је једнака димензији  $l_3$  умањеној за димензију  $c_2$ .

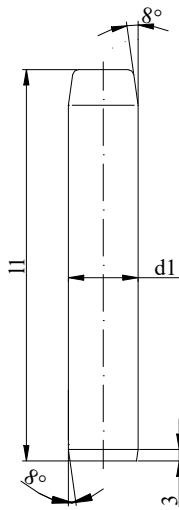
Ентитет 23 је дуж чија је почетна тачка једнака крајњој тачки ентитета 21; угао у односу на  $x$ -осу је 0 радијана; дужина дужи је једнака половини димензије  $d_3$ .

Ентитет 24 је кружни лук чија је  $x$  координата крајње тачке једнака  $x$  координати почетне тачке ентитета 1 умањена за половину димензије  $d_4$  и још умањена за половину димензије  $c_2$ , а  $y$  координата крајње тачке је једнака  $y$  координати почетне тачке ентитета 1;  $x$  координата почетне тачке је једнака  $x$  координати почетне тачке ентитета 1 умањеној за половину димензије  $d_4$  и још умањена за половину димензије  $c_2$ , а  $y$  координата почетне тачке је једнака  $y$  координати почетне тачке ентитета 1 умањена за димензију  $c_2$ ;  $x$  координата тачке центра је једнака  $x$  координати почетне тачке умањена за половину димензије  $d_4$  и још умањена за половину димензије  $c_2$ , а  $y$  координата тачке центра је једнака  $y$  координати почетне тачке ентитета 1 умањеној за половину димензије  $c_2$ .

Овим је моделирана лева страна машинског елемента чеп за алат, а командом за пресликавање може се пресликати на десну страну ова моделирана лева страна чепа.

### 6.3. Примена релационих исказа на једном фабричком типском елементу (стуб за алат)

Форма

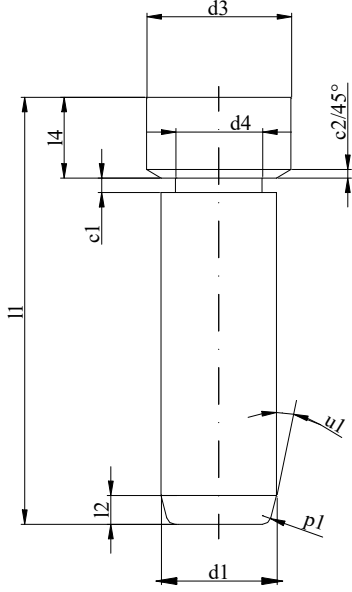


Слика 13:Стуб за алат

*Програм*

```
{1:l:[m=200,200;n=xm(1),ym(1)]}
{2:l:[m=m(1);a=0%;l=d1/2]}
{3:l:[m=m(1);a=pi;l=d1/2]}
{4:l:[m=n(2);xn=xm(4)-3*sin(8*pi/180);yn=ym(4)-3*cos(8*pi/180)]}
{5:l:[m=n(3);xn=xm(5)+3*sin(8*pi/180);yn=ym(5)-3*cos(8*pi/180)]}
{6:l:[m=n(4);n=n(5)]}
{7:l:[m=n(2);a=90%;l=l-3-8]}
{8:l:[m=n(7);xn=xm(8)-8*sin(8*pi/180);yn=ym(8)+8*cos(8*pi/180)]}
{9:l:[m=n(3);a=90%;l=l-3-8]}
{10:l:[m=n(9);xn=xm(10)+8*sin(8*pi/180);yn=ym(10)+8*cos(8*pi/180)]}
{11:l:[m=xn(9)+l,yn(9);N=xn(7)-l,yn(7)]}
{12:c:[IC(10);r=l.5;a=82%;xc(12)>(xm(12)+xn(12))/2;T=n(10);N=N(10)]}
{13:c:[IC(8);r=l.5;a=82%;yc(13)<(ym(13)+xn(13))/2;M=N(8)]}
{14:l:[m=M(12);N=n(13)]}
{15:l:[m=xs(14),ys(14)+2;n=xs(6),ys(6)-2]_..}
```

**6.4. Примена релационих исказа и фиктивних ентитета на једном фабричком типском елементу (стуб за алат)**

<p><i>Форма</i></p> 	<p><i>Програм</i></p> <pre>{1:l:[M=100,100;N=Xm(1),Ym(1)]} {2:l:[M=Xm(1),Ym(1)+l2;A=0%;L=(d1/2)]} {3:l:[M=N(2);A=90%;L=l-l2-l4-c1]} {4:l:[M=N(3);/(2);xn&lt;xm;L=(d1/2)]} {5:l:[M=Xm(1)+(d4/2),Ym(1)+l-l4-c1;-(2);yn&gt;ym;L=c1]} {6:l:[M=Xm(1),Ym(1)+l-l4;A=0%;L=(d3/2)-c1]} {7:WL:[M=N(6);A=45%;L=l4]} {8:l:[M=Xm(1),Ym(1)+l-l4+c2;A=0%;L=(d3/2)]} {9:l:[M=N(6);N=N(8)]} {10:l:[M=N(9);A=90%;L=l4-c1]} {11:l:[M=N(10);A=180%;L=(d3/2)]} {12:WL:[M=Xm(1),Ym(1);A=0%;L=l1]} {13:WL:[M=N(2);A=(270-u1)%;L=l1]} {14:WO:[C=Xm(1)+(d1/2)-(tg(u1*pi/180))*(l2-pl)-pl,Ym(1)+pl;R=p1]} {15:WL:[M=C(14);A=0%;L=l1]} {16:WL:[M=C(14);A=270%;L=l1]} {17:l:[M=M(1);N=P(16,l2)]} {18:l:[M=N(2);IC(14);A&gt;]} {19:c:[M=N(17);N=N(18);C=C(14)]}</pre>
--	--

Релациони исказ “нешто” мање од “нечега” дат је у командној линији 4 ( $xn < xm$  –  $x$  координата крајње тачке дужи је мања од  $x$  координате почетне тачке дужи).  
 Релациони исказ “нешто” веће од “нечега” дат је у командној линији 5 ( $yn > ym$  –  $y$  координата крајње тачке дужи је већа од  $y$  координате почетне тачке дужи).  
 Релациони исказ “нешто” веће дат је у командној линији 18 ( $A >$  – угао у односу на  $x$ -осу је већи).

## 6.5. Програмска реализација примарног описа кривих линија

(Приказ неких програмских сегмената који настају као резултат рада програмског модула за њихово генерисање)

Основа концепција рада програмског модула за генерисање сегмената за додељивање вредности карактеристичним тачкама, координатама карактеристичних тачака и другим параметрима изгледа као што је дато на слици 14. У оквиру механизма за анализу описа ентитета и синтезу програма врши се лексичка и синтаксичка анализа описа ентитета.

У даљем тексту овог рада се дају прикази програмских сегмената који настају као резултат рада програмског модула за њихово генерисање, чија је основна поставка дата шематски на слици 13. Ако се на улазу налази примарни опис криве линије као ентитета, као у следећем примеру:

```
x=x;y=20+2*sin(x);xm=20;xn=100
```

у излазу би се добио следећи скуп програма у *AutoLISP*-у:

```
(DEFUN WXQQ1()  
(setq XQQ1 (* (float 1) (float X)))  
)  
(defun WXQQ()  
(if(/= X nil)(progn)(progn(PRIHC " unesite X:"  
(setq X (atof(getstring))))))  
(terpri)  
(WXQQ1)  
(setq XQQ XQQ1)  
)  
(DEFUN WYQQ2()  
(setq YQQ2 (* (float 1) (float X)))  
)  
(defun WYQQ1()  
(WYQQ2)  
(setq YQQ1 YQQ2)  
)  
(defun WYQQ3()  
(WYQQ1)  
(setq YQQ3 (SIN YQQ1))  
)  
(DEFUN WYQQ4()  
(WYQQ3)  
(setq YQQ4 (* (float 2) (float YQQ3))))  
(DEFUN WYQQ5()  
(WYQQ4)  
(setq YQQ5 (+ (float 20) (float YQQ4))))  
(defun WYQQ()  
(if(/= X nil)(progn)(progn(PRIHC " unesite X:"  
(setq X (atof(getstring))))))  
(terpri)  
(WYQQ5)  
(setq YQQ YQQ5))
```



Слика 14: Модул за генерисање малих АутоLISP програма из командне линије за опис ентитета

Дакле, на основу улазног описа ентитета механизам за анализу описа ентитета и синтезу програма производи ове програме који се извршавају у пакету *AutoCAD*, чиме се добијају вредности координата криве линије као ентитета, на основу чега се може лако реализовати цртање криве линије у неком од *CAD* пакета.

## 6.6. Кратки опис језика за 2D параметарско геометријско моделирање и препроцесора за његово коришћење

Главна предност овог језика за моделирање лежи пре свега у расположивости свих могућих варијанти описа ентитета. Карактеристичне тачке, координате карактеристичних тачака и карактеристични параметри, као и искази који се користе у примарном (геометријском) опису, омогућују велики број варијанти за опис сваког стандардног ентитета (дуж, кружни лук, кружница). Опис геометрије ентитета може да се изведе и помоћу задатих функција у параметарском или непараметарском облику. Могу се описати форме са

конкретним димензијама (константама), општим димензијама (варијаблама) и комбинацијом конкретних и општих димензија.

Када се форме опишу, у овом језику за моделирање постоји препроцесор који опис форме (програм) процесира (са унетим конкретним димензијама пре процесирања или у току процесирања) и предаје у *AutoCAD*, где се добија конкретан цртеж.

Цртеж се добија са димензијама које су дефинисане у самом опису форме, а променљиве (опште) димензије се одабирају или уносе пре процесирања описа форме или се уносе у току процесирања форме (ако нису одабране или унете пре процесирања описа форме).

Код форми које имају само конкретне димензије, као што је форма названа *крива* у следећем примеру, позивом имена програма *крива*, добија се одмах цртеж форме *крива*.

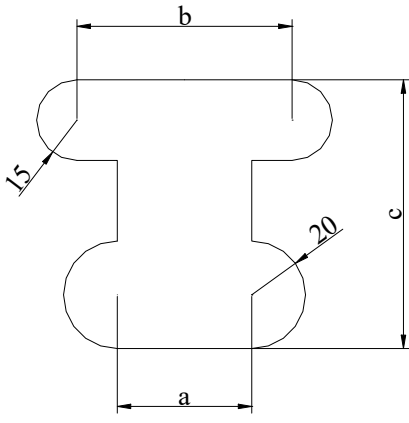
Машински елемент или друга форма која садржи променљиве димензије може да се опише у овом језику (то је у овом примеру форма Фиг1) и када се пожели да се елемент или форма нацрта са жељеним димензијама, једноставно се позове опис тог дела (програм за опис у овом примеру је Фиг1.txt), унесу димензије, после чега се добија излазни цртеж у *AutoCAD*-у.

Овде су за илустрацију дата два примера описа форме, као и излазни цртежи у *AutoCAD*-у.

Форме и примери описа форми, тј. програма у језику за 2D параметарско геометријско моделирање

<p><i>Форма:крива</i></p>
<p><i>Програм:крива</i></p> <pre>{1:F:x=x,y=20+2*sin(x)[xm=20;xn=100]} {2:F:x=x,y=(x-2.8)/(x*(x-30))[xm=40;xn=100]} {3:l:[M=M(2);N=xm(1),ym(2)]} {4:l:[M=n(1);n=n(2)]} {5:wl:[m=m(1);N=n(3)]} {6:o:[c=80,40;r=4]} {7:f:y=ys(5)+(L(5)/2)*sin(t);x=xs(5)+(L(5)/2)*cos(t)[tm=pi/2;tn=3*pi/2]} {8:o:[C=xs(5),ys(5);R=R(6)*2]}</pre>



<p>Форма: Фиг1</p> 	<p>Програм: Фиг1.тхт</p> <pre>{1:l:[m=100,100;n=xm(1),ym(1)]} {2:l:[m=m(1);a=0;l=a/2]} {3:c:[c=xm(1)+(a/2),ym(1)+20;m=n(2);a=180%]} {4:l:[m=n(3);l=c-70;a=90%]} {5:l:[m=n(4);a=0;l=(b/2)-(a/2)]} {6:c:[c=xn(5),yn(5)+15;m=n(5);a=180%]} {7:l:[m=n(6);50a=180%;l=b/2]} {*;MM:2:3:4:5:6:7[m=m(1);n=n(7)]}</pre>
--	---

### Коришћење препроцесора

**I.** Потребно је копирати све датотеке описане у овом језику (у овом случају то су фиг1.тхт и крива) и препроцесор са дискете, диска и сл. на диск с: и то на омотницу (директоријум) *Acad*;

**II.** Покренути *AutoCAD*.

1.

Унети у командној линији

Command:(load"PM")

2.

Унети у командној линији и поткомандама

Command:PRES

Specify data model:kriva

Delete objects? [Yes/No]< No >:Y

3.

Унети у командној линији и поткомандама

Command:PRES

Specify data model:fig1.txt

Specify a:30

Specify b:50

Specify c:100

Delete objects? [Yes/No]< No >:Y

4.

Унети у командној линији и поткомандама

Command:PRES

Specify data model:fig1.txt

Specify a:**30**  
Specify b:**70**  
Specify c:**120**  
Delete objects? [Yes/No]< No >:Y

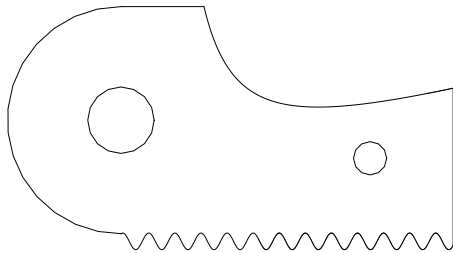
5.  
Унети у командној линији и поткомандама

Command:**PRES**  
Specify data model:**fig1.txt**  
Specify a:**40**  
Specify b:**240**  
Specify c:**200**  
Delete objects? [Yes/No]< No >:Y

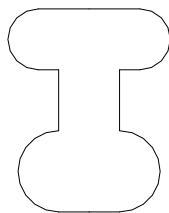
Напомена: Све што је болдирано (написано тзв. масним словима) и иза знака :  
(двотачка) уноси се са тастатуре.

Резултати при коришћењу препроцесора (команди описаних на претходним  
примерима: крива и фиг1.тхт) дати су на овој и следећој страни:

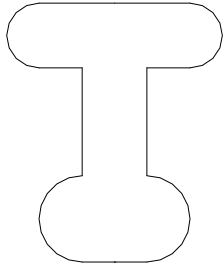
Крива (резултат корака 2 из упутства)



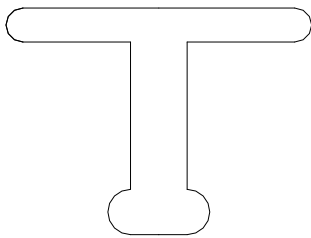
Фиг1.тхт (резултат корака 3 из упутства)



Фиг1.тхт (результат корака 4 из упутства)



Фиг1.тхт (результат корака 5 из упутства)



### **6.7. Моделирање (друга фаза) за издвајање једног од два скупа тачака (једне колекције геометријских података) када се користе релациони искази**

Када се на основу улазних геометријских података као што су карактеристичне тачке, координате карактеристичних тачака, релација паралелно, управно и тангентно и геометријских параметара као што су дужина, угао, полупречник, пречник и слично не може доћи до једног решења, тада се у даљи ток поступка уводе релациони искази. То су релациони искази типа: “нешто” веће од “нечега” или “нешто” мање од “нечега” који садрже релацију веће (од) или релацију мање (од) или релацију која бар садржи релацију веће (од) или релацију мање (од), као што су релације веће или једнако или мање или једнако, није веће (од) или није мање (од), није веће или једнако или није мање или једнако, и где је у релациони исказ бар на једној страни укључена бар једна координата карактеристичне тачке, као што је почетна тачка, крајња тачка, централна тачка и слично, или бар један карактеристични геометријски параметар, као што је угао, дужина, пречник, полупречник и сл. из ентитета који се моделира, и то она координата оне карактеристичне тачке или онај карактеристични геометријски параметар који има два решења на основу улазних геометријских података.

У поступку (фаза 2) за добијање једног од два решења за ентитет који има поступак моделирања такав да је у њега укључен релациони исказ овог типа, ток је следећи:

Укључи релациони исказ. Ако је потребно, израчунај аритметички израз лево и десно од дате релације веће или мање.

Упореди вредности са обе стране релације и одреди да ли је вредност релационог исказа 'истина' или 'није истина'.

Ако је код првог извршења (операције, команде, командне линије и слично) вредност релационог исказа 'истина', све вредности за координате карактеристичних тачака (и карактеристичних геометријских параметара) којим је одређен ентитет остају исте, а ако је вредност релационог исказа 'није истина', тада се мењају вредности координата карактеристичних тачака и других карактеристичних геометријских параметара (преузима се друга колекција података, као што су карактеристичне тачке, координате карактеристичних тачака и параметри, као што су угао, дужина, пречник, полупречник), тако да вредност релационог исказа буде 'истина'. Може и после преузимања другог решења да се врши поређење са подацима из тог решења да ли је вредност релационог исказа истина.

Другим речима, преузима се оно решење чији геометријски подаци у релационом исказу доводе до истинитости релационог исказа.

Тако се издваја скуп тачака за жељени ентитет на основу постављеног захтева да вредност релационог исказа буде 'истина'.

Овде се пореде вредности са леве и десне стране релације.

У поступку (фаза 2) за добијање једног од два решења за ентитет који има поступак моделирања такав да је у њега укључен релациони исказ типа “Нешто” веће или “Нешто” мање и слично томе (у “Нешто” је укључена бар једна координата карактеристичне тачке или бар један карактеристични геометријски параметар, и то она координата или онај карактеристични параметар ентитета који се моделира који има два решења на основу осталих специфицираних геометријских података и/или релација или је у “Нешто” укључена дистанца између специфицираних тачака, при чему у том случају једна од тих специфицираних тачака је тачка која има два решења настала у претходном делу поступка), ток поступка је следећи:

Укључи релациони исказ. Ако је потребно, израчунај аритметички израз са једне стране дате релације веће односно мање, и то са подацима из оба решења. Упореди вредности из оба решења и издвој оно решење за које је вредност релационог исказа 'истина'. Другим речима, преузима се оно решење чији геометријски подаци у релационом исказу доводе до истинитости релационог исказа. Овим се поступком издваја скуп тачака за жељени ентитет на основу постављеног захтева да вредност релационог исказа буде 'истина'.

Овде се у поступку моделирања пореде вредности из оба решења за ентитет који се моделира.

У поступку (фаза 2) за добијање једног од четири или осам решења, за ентитет који има поступак моделирања такав да је у њега укључен релациони исказ типа “Нешто” највеће или “Нешто” најмање и слично томе, ток је следећи:

Укључи релациони исказ. Упореди вредности из свих решења и издвој оно решење за које је вредност релационог исказа 'истина'. Другим речима, преузима се оно решење чији геометријски подаци у релационом исказу доводе до истинитости релационог исказа. Овим се поступком издваја скуп тачака за

жељени ентитет на основу постављеног захтева да вредност релационог исказа буде 'истина'.

Овде се у поступку моделирања пореде вредности из свих решења за ентитет који се моделира.

У поступку (фаза 2) за добијање двоструко мање решења, за ентитет који има поступак моделирања такав да је у њега укључен релациони исказ типа “Нешто” ближе “Нечему” или “Нешто” даље од “Нечега” и слично томе, ток је следећи:

Укључи релациони исказ. Израчунај растојање између тачака специфицираних у релационом исказу, и то са подацима из свих решења. Упореди вредности из свих решења и издвој она решења за које је вредност релационог исказа 'истина'. Другим речима, преузимају се она решења чији геометријски подаци у релационом исказу доводе до истинитости релационог исказа. Овим се поступком издваја скуп тачака за жељени ентитет на основу постављеног захтева да вредност релационог исказа буде 'истина'.

Овде се у поступку моделирања пореде вредности из свих решења за ентитет који се моделира.

У поступку (фаза 2) за добијање једног од четири или осам решења, за ентитет који има поступак моделирања такав да је у њега укључен релациони исказ типа “Нешто” најближе “Нечему” или “Нешто” најдаље од “Нечега” и слично томе, ток је следећи:

Укључи релациони исказ. Упореди вредности из свих решења и издвој оно решење за које је вредност релационог исказа 'истина'. Другим речима, преузима се оно решење чији геометријски подаци у релационом исказу доводе до истинитости релационог исказа. Овим се поступком издваја скуп тачака за жељени ентитет на основу постављеног захтева да вредност релационог исказа буде 'истина'.

Овде се у поступку моделирања пореде вредности из свих решења за ентитет који се моделира.

Издавање решења за ентитет који се моделира у алгоритамским и програмским извођењима изводи се тако да се преузима (издваја) оно решење из којег подаци доводе до истинитости (тачности) релационог исказа.

## 7. ПРОЈЕКТАНТСКА БАЗА ПОДАКА КАО ПОДРШКА ЈЕЗИКУ ЗА 2D ПАРАМЕТАРСКО ГЕОМЕТРИЈСКО МОДЕЛИРАЊЕ

Пројектант код пројектовања алата, прибора и сличних машинских конструкција користи разне књиге и приручнике, итд. Осим тога, он укључује и своје писане или неписане личне искуствене информације. Стога се овде од тога пошло и циљ је да се све ово интегрише, односно представи и користи у што природнијем облику. Развијен је ради тога један део система који служи као помоћ пројектанту, а у њему су интегрисане све врсте информација. Дакле, све врсте информација (константе, формуле, дијаграми и разни други графови, текстови обавештења и објашњења) из разних књига, приручника, каталога, као и информације из искуства појединог пројектанта или групе пројектаната потребне за пројектовање спајају се у једну целину. Као такве се и користе. Пројектант овде уместо разних информација датих на уобичајен начин у књигама, приручницима или белешкама користи рачунар. Могло би да се каже да користи електронску књигу, само што је знатно лакше листање и притом директније коришћење у раду. Једино обавештења и објашњења се не користе директно, већ на исти начин као из уобичајених извора информација код класичног пројектовања. Овде су обавештење и објашњења све оно у информационој бази што није формула, граф или други важан податак који се директно користи код пројектовања алата или сличног машинског склопа или елемента, односно све информације које нису предмет транспорта и трансформације код пројектовања. Дакле, структура података у пројектантској „електронској” књизи је састављена из једног или више елемената следеће колекције [58,59]:

S1, S2, S3, S4, S5, S6.

Овде се поједини подаци из горње колекције могу поделити на оне које су предмет транспорта и трансформације код пројектовања (примарни подаци):

- S1 – формуле,
- S2 – графикони (дијаграми),
- S3 – графички ентитети – опис модела (пројекције, погледи, пресеци итд.),
- S4 – табеле и константе,  
са посебном додатном подгрупом у оквиру примарних података
- S5 – извршиви програми,  
и податке информативног карактера који нису предвиђени за транспорт  
и трансформацију (секундарни подаци),
- S6 – обавештења и објашњења.

Формуле се пишу на уобичајен начин, с тим што се неки знакови прилагођавају скупу знакова на улазним уређајима рачунара. Линије у графиконима (дијаграмима) могу да се замене описом неког од основних ентитета (аналитички описивих или аналитички неописивих).

Графички ентитети – графички описи, тј. модели објеката (пројекције, погледи итд.) могу да буду:

- модели са константном формом и константним димензијама S3a,
- модели са константном формом и променљивим димензијама S3b,
- модели комбиновани од S3a и S3b (са константном формом и комбинацијом константних и променљивих димензија),
- модели са променљивом формом и константним димензијама S3d,
- модели са променљивом формом и променљивим димензијама S3e,
- комбиновани од S3d и S3e (са променљивом формом и комбинацијом константних и променљивих димензија).

Подаци у табелама и друге константе могу да буду нумерички, алфанумерички или било какви други низови знакова.

Извршиви програми могу да служе за прорачун, моделирање (цртање) и опис неке криве на графикону (дијаграму), претраживање у базама података итд.

Обавештења и објашњења могу да буду текстуалног или графичког типа.

Скоро сви од наведених података, изузев делимично S5, могу да се налазе у самој књизи. Међутим, ради прегледности и што природније структуре података у књизи, неки од ових података се смештају ван ње, а у књизи остаје само њихова идентификација на основу које се успоставља веза између податка датог том идентификацијом и дела меморије у којем налази опис неког елемента из структуре података.

Већина од ових података могу да се дају у изворном облику (као и тамо одакле су преузети) или са минималним изменама. Наравно, могућа је реорганизација са циљем да се избаци што више информација које нису неопходне за пројектовање. Циљ је да ова електронска књига буде што концизнија и због тога што је мањи број информација прегледнији за пројектанта и, са друге стране, да се не заузима непотребно меморија рачунара.

Зависно од организације, подаци се могу сместити у једну књигу са више поглавља или у више књига или у више књига са више поглавља.

Разни системи за пројектовање обично нису довољно флексибилни и стога се овде одмах ишло на то да се ничим не спутава слобода пројектанта да уноси, чита и користи информације које жели, и то на један сличан начин као што их иначе користи код класичног пројектовања.

Формирање ове пројектантске књиге не разликује се много од уобичајеног писања књиге, приручника или каталога. Иначе, овде се пошло и од тога да се комплетни каталози појединих произвођача елемената или полуфабриката пренесу у пројектантску књигу у изворном облику, чиме се онеме који формира ту књигу максимално олакшава посао. Такође, и комплетни делови неких књига или приручника могу се пренети у пројектантску књигу. Наравно, овде је потребно извршити у неким случајевима редукцију такву да се избаце непотребне речи и реченице, односно да у књизи остану само употребљиве информације, било да су то информације које се директно користе, у смислу да се са њима врши манипулација или трансформација, а посебно да се изврши редукција података информативног карактера (обавештења, објашњења и слично.). Као у свакој књизи, на првим странама је садржај на коме се налазе називи поглавља и делова поглавља и стране од којих почињу та поглавља односно делови поглавља (слика 14).

Ако се, на пример, жели одмах прећи на одређено поглавље, није неопходно да се листа страна по страна, мада и тако може, него се корисник може одмах пребацити на тражено поглавље избором тог поглавља у садржају. Свако поглавље се пише као и у класичним књигама, приручницима, каталозима итд. Код писања константи и формула и др., односно информација са којима се врши манипулација и трансформација, код самог пројектовања се користи обележавање маркерима (мада и није неопходно) ради лакшег уочавања тих података.

---

Sadržaj

Dodatak oko konture dela na pripreмку (traci).....	1
Vijci za vodjenje opruga.....	7
Proracun probojaca.....	9
Caure za vodjenje.....	11
Stubovi za vodjenje.....	12
Tehnoloski zazor.....	13

Слика 15: Пример садржаја пројектантске базе података

Из следећег примера дела текста у поглављу о опругама (слика 15) може се видети како то практично у некој од варијанти може да изгледа.

---

**Maksimalna dozvoljena sila kojom se sme zavojna opruga okruglog preseka opteretiti odredjuje se po formuli:**

$$F_{max} = 3.1457 \cdot (d \sim 3) \cdot T_{ud} \text{ [daN]} \dots \dots \dots (66)$$

gde je d [mm] - prečnik žice,  
 $T_{ud}$  [daN/mm<sup>2</sup>] - dozvoljeno naprezanje na uvijanje, a zavisi od vrste materijala (tabela 77)  
D [mm] - prečnik opruge.

Ovde se za prečnik žice d uzimaju vrednosti '1', '1.5', '2', '2.5', itd.

d='2.0'	(67)
D='10.0'	(68)

---

Слика 16: Део поглавља о опругама у пројектантској бази података



Нешто се разликује писање формула од уобичајеног. Писање формула и других података ове књиге се прилагођава знаковима на тастатури рачунара.

Уместо формуле дате у примеру на слици 15 дела текста о опругама може се унети и нека друга формула у току рада, односно било који аритметички израз, укључујући ту и константу.

Придруживање константе код овог примера је дато на примеру пречника који треба да се унесе да би се могла израчунати вредност за силу  $F_{max}$ . Вредност пречника  $d$  може да се унесе на два места – избором или уносом вредности на месту

'1', '1.5', '2', '2.5',

или на месту

$d=2.0$ ,

где се овај пречник добија из формуле 67. За пречник  $D$  могао је уместо избора константе '10.0' на овом месту да се унесе и било какав други аритметички израз. На пример, могао је да се унесе аритметички израз на месту '10.0' (уместо овог податка)  $5*d$  и добио би се исти резултат.

Константе, као што се види са ове слике, пишу се на уобичајен начин. Осим тога, константе се могу налазити, на пример, и у некој табели у неком поглављу на некој страни пројектантске књиге и она, између осталог, може да се пише и на начин дат на слици 16.

**табела 44**

Пречник опруге D	Пречник жице d
'16'	'1.4'
'16'	'1.6'
'16'	'2.0'
'18'	'2.5'
'22'	'2.5'
'22'	'3.0'

Слика 17: Подаци за моделирање у табели

Линије у разним дијаграмима се могу апроксимирати правим или кривим линијама (аналитички описивим или аналитички неописивим). Тако би се могли делови дијаграма који изражавају неку зависност апроксимирати једначинама, као у примеру датом на слици 17, где се на месту параметра уноси вредност параметра, а резултат се добија преко аритметичког израза којим се апроксимира нека зависност на дијаграму. Уместо аритметичког израза могу се, ако је потребно, дати и скупови тачака којим се описује нека линија на дијаграму и апроксимација као додатни део описа такве линије на дијаграму.

Могу се код рада са дијаграмима укључити и извршиви програми ако је потребно.

---

	$x$ =интервал параметра са дијаграма	
1-	$y$ =ариметички израз 1 .....	(71)
2-	$y$ =аритметички израз 2 .....	(72)
3-	$y$ =аритметички израз 3 .....	(73)

---

Слика 18: Апроксимација дијаграма једначином

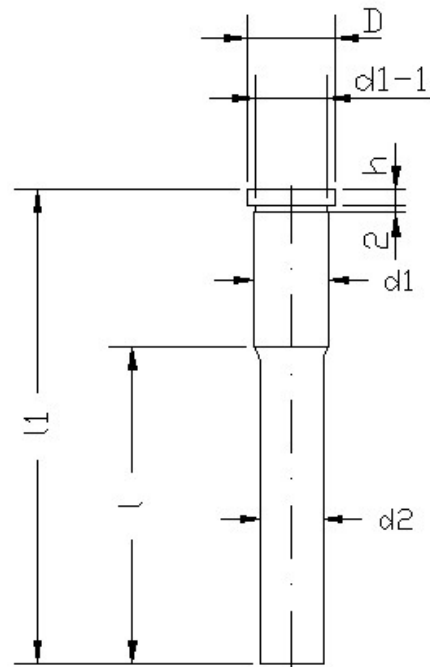
Значајан део у већини техничких књига, приручника, каталога и сл. везаних за пројектовање јесу графови који дају информације о облику сложених графичких ентитета. Овде се под сложеним графичким ентитетом подразумевају, на пример, у машинству подсклопови и елементи и њихове пројекције, пресеци и детаљи. За опис оваквих сложених ентитета развијен је језик за 2Д параметарско геометријско моделирање.

Сваки од набројаних сложених ентитета се посебно описује овим језиком и тај опис се придружује идентификацији ентитета датај у књизи. У самој књизи се само приказује облик који се добија цртањем са једним низом димензија из мноштва константи за те димензије. На пример, слика једне пројекције једног машинског елемента може да се презентује у пројектантској књизи на уобичајен начин, само што се идентификација слике пише између маркера (слика 18) [46,47]. У ствари, овде се разликују два типа графова. Постоје графови који служе само да дају обавештење о облику и саставу овог или оног објекта, и претходно поменути графови, чија се изградња са конкретним димензијама очекује у току самог пројектовања.

Код првог типа графова не постоји потреба за посебном идентификацијом графа осим броја слике у оквиру књиге или у оквиру једног поглавља, зависно од тога како је уређена идентификација слике. Код оваквих графова даје се идентификација испод или поред слике исто као и код класичног обележавања графова. Међутим, ако се очекује изградња модела тог графа, у смислу да он добија стварне димензије у току процеса пројектовања, тада се идентификација слике ставља између маркера, односно требало би да постоји неко обележје које одмах упућује на то о којем се типу графова ради.

Ова идентификација може да буде и назив датотеке у којој се налази опис тог графа. Ово је један елемент из колекције који може да се налази изван саме књиге, а директно се користи у раду.

Могу се моделирати сложени графички ентитети и у другим језицима. Тада је идентификација слике и име датотеке програма за опис тог сложеног ентитета, или је име програма у некој вези са именом слике.



'sl-5'

Слика 19: Презентовање моделираног пробојца у пројектантској бази података

Иначе, у било ком делу књиге дозвољено је уношење извршивих програма за опис сложеног графичког ентитета, опис неке криве у дијаграму или неки прорачун за који није неопходно или није могуће да се пише у самој пројектантској књизи. Уместо избора неке слике може да се унесе и идентификација неке друге слике која се не налази у књизи. Тада се морају посебно уносити димензије потребне за цртање датог графа.

У целини, овај део система треба да омогућује и лако почетно формирање пројектантске књиге, али и касније лаку доградњу и измене у њој. На исти начин се врше допуне књиге, као и када се она први пут формира, што омогућује да сваки пројектант формира прво једну иницијалну књигу, а касније је лако проширује или врши исправке у њој. Ово важи не само за директне допуне и измене у самој пројектантској књизи него и код допуна и измена у опису сложених графичких ентитета језиком за  $2D$  параметарско геометријско моделирање развијеном као подршка овом концепту. Дакле, циљ код овог концепта је био, с једне стране, да се интегришу све врсте података, и то подаци који се директно користе код пројектовања (константе, формуле, дијаграми и графови разних сложених графичких ентитета) и евентуално извршиви програми, као и подаци који се користе индиректно (обавештења, објашњења и сл.), а са друге стране, да се интегришу различити типови података (нумерички,

алфанумерички, низови знакова и графички подаци). Осим тога, могу се описи формулама претварати у графичке описе и обрнуто – графички описи у описе формулама, као што је то код дијаграма. Исто тако неки подаци могу да се дају као алфанумеричке константе, а да се код коришћења у току пројектовања трансформишу у нумеричке податке. На пример, могу се дати овакве константе у тексту неког поглавља ... 'аутоматско' ... 'ручно' ..., где из избора једне од ових константи произилази избор неког коефицијента који добија касније нумеричку вредност потребну за прорачун у некој формули или за цртање неког сложеног графичког ентитета.

Ако се анализирају саставни делови процеса пројектовања може да се закључи да су то читање обавештења и објашњења, прорачуни, цртање, димензионисање, избор ових или оних параметара и коефицијената. Све те активности, уз коришћење овакве (електронске) књиге код пројектовања, могу да се убрзају и да омогуће пројектанту угоднији рад.

Притом се посебно олакшава рад код транспорта и трансформација информација. Овде се у пројектантској (електронској) књизи налазе базни подаци, састављени од елемента из колекције S1, S2, ..., S6 потребни за неко подручје примене.

Базне информације се смештају у пројектантску (електронску) књигу и, ако је потребно, са додатним скуповима информација везаним за неку идентификацију у самој књизи, а други део информација, према овом концепту информационог система се смешта у „радну датотеку”. У овај други део информација спада улазне информације и информације добијене у току пројектовања. Дакле, овде постоје две главне целине. То су пројектантска књига и радна датотека. Радна датотека се састоји од два дела. У првом делу се смештају сви међурезултати, односно сви резултати настали током пројектовања неког објекта. Подаци који се смештају у овај део радне свеске нису графичког већ текстуалног типа.

На пример, овде се смештају изабране или унесене вредности ових или оних константи, резултати израчунавања помоћу формула, резултати добијени помоћу дијаграма или подаци о избору неког сложеног графичког ентитета. Овде сваки од ентитета, као резултат рада при пројектовању, има свој опис. Ту спада идентификација у оквиру целог система и стварно име неке променљиве која добија вредност придруживањем вредности неке константе, израчаног аритметичког израза из формуле или неке слике одабране из пројектантске књиге, или додељивањем неког другог скупа података из колекције података (S1, S2, S3, итд.) Осим тога, сваки ентитет у овом делу радне свеске има и додатни опис. То је, на пример, информација о страни у пројектантској књизи, поглављу и самој књизи из које је тај ентитет одабран или унесен, односно информација о положају (месту) ентитета. Ако се ради о податку из табеле, онда се даје и број табеле, или ако се ради о формули, онда се даје и број формуле у оквиру књиге или поглавља. Ту се налази и додатни кратки опис ентитета.

Други део је сама радна документација графичког типа о објекту који се пројектује. Овде се налазе за сваки ентитет (склоп, подсклоп, елемент) уз графички опис геометрије (довољан број пројекција, погледа итд.) и остали подаци потребни за детаљан опис ових ентитета. Код склопног цртежа то су идентификација саставних компоненти склопа у табели, тј. саставници склопа. Даље, у овом делу радне свеске за компоненте склопа се дају, осим основних

графичких ентитета којим се у основи описује свака компонента, коте и знакови обраде, а ако је потребно, и опис међусобног положаја појединих површина. Притом се као допунски опис дају и допунска објашњења и обавештења везана за израду и монтажу компоненти или склопа у целини.

У посебном делу радне свеске се смештају и међурезултати графичког типа. То су делови пројектне документације који се добијају у току изградње целине графичког описа разних компоненти и склопа. Код пројектовања објекта, сви међурезултати почев од првог налазе се у овом делу радне свеске.

Сви ови сложени графички ентитети у радној датотеци имају своју глобалну и локалну идентификацију: глобалну на нивоу система и локалну у оквиру компоненте или склопа који се компонује. Тако се са њима може вршити манипулација или доградња ове или оне компоненте.

Димензије потребне за цртање неког објекта могу да се добију:

- а) као резултат избора или уноса једне или више константи (из неке табеле или једног, односно више места у пројектантској књизи),
- б) као резултат прорачуна из једне или више формула одабраних или унетих на месту одговарајућих формула,
- в) као резултат који произилази из једног или више дијаграма,
- г) као резултат рада неког извршивог програма,
- д) комбинације а–г.

Могуће је комбиновати добијање неког од резултата избором или уносом (код цртања, прорачуна формулама, са дијаграма, из табеле, из извршивог програма) на све могуће начине.

У пројектантску (електронску) књигу се могу сместити подаци који имају исто значење из разних литературних, искуствених и других извора. Неки податак потребан у неком тренутку за даљи рад се из ове књиге може добити на више начина. Неки податак се може добити из било које од неколико табела, или као резултат израчунавања из више формула, или из више графикона, или из једног од више извршивих програма.

Дакле, неки тренутно потребан податак се може добити на разне начине и са разних места (положаја) у оквиру неког дела простора базних информација.

У примеру датом у додатку А као делу неке електронске пројектантске књиге се може један податак (б) узети из табеле 1 или табеле 2, или прорачуном из формуле 5, односно формуле 8.

Једна од варијанти лексичке структуре дата је у следећем поглављу ради даљег објашњења концепта овог подсистема (рада са пројектантском књигом).

#### *Информациона база (књига – радна датотека)*

У наредном делу текста објашњен је детаљан рад система за пројектовање, и то у делу који се односи на транспорт и трансформацију потребних података из пројектантске (експерт) књиге у радну свеску, а који произилази из претходно постављеног концепта.

У овом делу ће се дати једна од могућих варијанти развијена да се покаже могућност оваквог начина рада.

Скуп речи који се користе у раду је минималан, тако да се пројектант не замара у раду са речима које користе, већ се може концентрисати на само пројектовање.

Подсистем за рад са књигом је такав да омогућује врло једноставно руковање са њом, почев од формирања књиге, евентуалних измена и допуна, до избора или уноса неког податка потребног за пројектовање.

У лексичкој структури користе се следећи скупови знакова:

- велика и мала слова енглеског алфабета: A, B, C, ..., Z.
- знаменке: 0, 1, ..., 9.
- остали знакови: ., +, -, <, \*, /, ~, (,), [, ], {, }, @, бланко и други знакови са улазних уређаја.

У речнику, у оквиру лексичке структуре, према намени, постоје две врсте речи:

- речи које се користе у књизи, тј. простору (равни) података, као што су константе, формула итд.,
- речи које производе неку акцију у оквиру књиге или између књиге и радне свеске, односно чији је резултат транспорт и трансформација, односно манипулација подацима.

Прва врста речи су следеће класе речи у речнику:

- све речи које се користе у неком језику,
  - константе (алфанумеричке константе, цели и реални бројеви и специјална константа  $\pi$ , низови знакова),
  - стандардне функције:  $\sin$ ,  $\cos$ ,  $\text{tg}$ ,  $\text{ctg}$ ,  $\text{arctg}$ ,  $\log$ ,  $\ln$  итд.,
  - идентификатори,
  - имена променљивих могу да се састоје од слова и знаменки, при чему на првом месту мора бити слово,
  - имена слика,
  - имена извршивих програма,
  - заграде  $\{\}()$ ,
  - остале речи.
- Под осталим речима подразумевају се речи које не припадају ниједној од горе наведених класа, а то су:
- релације  $>$ ,  $<$ ,  $>=$ ,  $<=$ ,
  - једнакост  $=$
  - операције  $+$ ,  $-$ ,  $*$ ,  $/$ ,  $\sim$   
( $*$  – ознака за множење), ( $/$  – ознака за дељење),  
( $\sim$  – ознака за степен)
  - ознака за степене код углова  $\%$
  - и друге остале речи као што су маркери  $^$ ,  $'$  итд.

Ова врста речи неће се посебно даље разматрати, јер је значење и комбиновање ових речи уобичајено.

## Садржај

Додатак око контуре дела на припремку (траци).....	1
Пресе.....	7<
Вијци за вођење опруга.....	17
Прорачун пробојаца.....	29
Чауре за вођење.....	31
Стубови за вођење.....	42
...	
...	
...	
Технолошки зазор.....	109

Слика 20: Пример преласка на поглавље у пројектантској бази података

У оквиру друге врсте речи налазе се специјалне речи:

<, +, -, \_ или ред, \$ или страна, I или избор, >> или излаз, ^, @.

Даље се даје значење и начин коришћења речи које производе неку акцију у оквиру пројектантске (експерт) књиге или између књиге и радне свеске.

- Реч за улаз на почетак неког поглавља <.

Користи се само у делу пројектантске књиге где се налази садржај. Означава акцију преласка на почетак означеног поглавља. На пример, ако једна страна на којој се налази садржај изгледа као што је дато на слици 19, онда ова реч значи прелазак на прву страну поглавља Пресе, односно прелазак на страну 7 пројектантске књиге.

Ако се поред ове речи дода идентификација (број) неке стране, онда се одмах прелази на ту страну са било које стране књиге (поглавља).

- Речи за листање страна и редова + и -.

Означавају акцију преласка са *i*-тог листа/реда на (*i-k*)-ти или (*i+k*)-ти лист/ред, где је *k* број знакова - или +. Ако је то *k* знакова -, изводи се акција преласка за *k* листова/редова уназад, а ако је то *k* знакова +, изводи се акција преласка за *k* листова/редова напред. На пример, ако се жели прећи са друге стране на четврту страну, онда команда дата у додатку А (на другој страни пројектантске књиге) производи акцију преласка на четврту страну.

Слично је и код листања редова. Тада се команда састављена од речи + или - користи за пребацивање по редовима у оквиру једне стране.

Може се после речи + додати и број за који се жели прескочити одређени број страна напред. На пример, команда +10 значи акцију преласка за десет страна напред. Слично томе, могуће је ако се уз реч - дода неки број, извести акцију преласка за тај број страна уназад.

На исти начин се могу брзо листати и редови у оквиру једне стране. Овде се може напоменути да се листање пројектантске (експерт) књиге може обављати и преко кључних речи. Тада би се аутоматски отварала прва, а користећи реч + и следеће стране на којима се налази та кључна реч. Набрајањем листе кључних речи извршила би се додатна редукција могућих страна на којима се оне налазе.

- Реч за прелаз на листање редова \_ или ред.

Означава да се жели листање редова у оквиру једне стране пројектантске књиге.

- Реч за прелаз на листање страна \$ или страна.

Означава да се жели листање страна пројектантске (експерт) књиге.

- Реч за избор или унос података i или I или избор.

Означава команду да се жели изабрати или унети скуп података из тог реда.

- Реч за прескакање једног знака у реду @.

Команда састављена од k ових речи значи да се не жели ништа изабрати или унети из тог дела реда, из кога се иначе жели одабрати или испод којег се жели унети скуп података.

- Реч за избор ^.

Ова реч, ако се унесе испод првог знака i-тог реда, означава да се желе изабрати све константе, формуле итд. које се налазе у том реду, а спадају у онај део података са којима је предвиђена манипулација. Сви такви подаци коришћењем ове речи ће се пренети у онај део радне свеске где се смештају међурезултати код пројектовања.

Погодан је на овај начин избор скупа података из табела. Ако се налазимо на неком делу стране где се налази нека табела (на пример страна 33 у Додатку А) и унесе се реч ^, то означава команду за избор константи, односно вредности за променљиве d1, d3 и d4 итд. којима се придружују вредности

d1 <- 18,

d3 <- 36,

d4 <- 52,

...

k <- 18.

Ове променљиве ће се сместити у радну датотеку са додатним описом.

Команда за појединачни избор неке константе, формуле или слике комбинује се од речи за избор ^ и речи за прескок @. Ако се жели изабрати само нека константа, формула итд., тада би се то извело командом састављеном од скупа речи за избор и скупа речи за прескакање. У примеру датом у Додатку А на страни 5 (табела 2, наставак) ова команда би имала дејство да придружи променљивој a1 вредност константе која се налази у датом реду (a1 <- 2.2). Може се на овај начин комбиновањем скупова речи ^ и скупова речи @ изабрати и више од једног податка из једног реда. Иначе, осим избора података из пројектантске књиге, дозвољен је и унос другог податка уместо одређене константе, формуле или слике. Тада се користи комбинација скупа речи за прескок и унос жељене вредности за ту константу (цели или реални број) или аритметички израз на местима где се налазе формуле. На пример, ако се жели унети вредност 2.1 уместо вредности 2.8 за константу која се придружује променљивој b, онда се то изводи на начин приказан у Додатку А на страни 4. Може се, слично као и код избора, унети и више константи или аритметичких израза чија се вредност, унесена или израчуната, придружује променљивој на датом месту у овину изабраног реда на некој страни.



На пример, у Додатку А на страни 34 је дат пример комбинованог избора и уноса: за d3 се уноси вредност 115, за d5 се уноси вредност 98, за d6 се уноси вредност 11.5, а за остале променљиве се врши избор из задњег реда табеле 15.

Осим избора са тастатуре, избор константи из разних табела и избор неке слике може се још лакше извести помоћу миша. Овакав начин избора константи или слика илустрован је на страни 11. На страни 11 бира се мишем притиском на знак <- колекција димензија из табеле 4 и приказује листа изабраних димензија, а после тога се бира слика чауре (сл. 6) притиском миша на идентификацију ове слике. После избора слике добија се чаура са изабраним димензијама.

Код избора слике или неке формуле неког извршивог програма и слично из радне датотеке узимају се раније одабрани, унесени или израчунати подаци потребни за цртање слике, неке формуле или подаци потребни за рад у неком извршивом програму. Ако их нема у радној датотеци, тада се тражи да се они унесу.

- реч за излаз из књиге >>.

Ова реч означава акцију изласка из пројектантске књиге или из једног дела уколико су све базне информације подељене на више делова.

Дакле, као што је изложено, све ове речи и команде служе за транспорт и трансформацију података (одабирање података или унос података који се додељују некој променљивој, активирање неке формуле одабране из књиге или унете на одговарајућем месту за израчунавање вредности неке променљиве, активирање неког извршивог програма и слично).

У току пројектовања одвија се интензивни трансфер података између пројектантске књиге и радне датотеке.

Осим идентификације ентитета (неког податка), у једном делу радне датотеке постоји увек одговарајући допунски опис ентитета који се преноси у њу.

Овај део целог система за пројектовање неког објекта омогућује избор или унос неког податка или подскупа из скупа података потребних за пројектовање неограничен број пута. Дакле, ако пројектант није задовољан оним што је раније одабрао или унео, може се поново вратити избору или уносу тог податка или подскупа података везаног за пројектовање алата или сличног машинског дела. База података смештена у пројектантску књигу треба да је таква да омогућава избор свих података релевантних за пројектовање једног објекта.

## ДОДАТАК А

strana 1

### ДОДАТАК ОКО КОНТУРЕ ДЕЛА НА ПРИПРЕMKU (TRACI)

Veličina odstojanja konture dela najblize ivici trake i rastojanje između kontura dva uzastopna dela na traci zavise najviše od debljine lima (s), velicine izratka odnosno širine trake i oblika dela koji se izrađuje. Prema Romanovskom ovaj dodatak (sl-1) se daje u zavisnosti od debljine lima i oblika dela koji se izrađuje (tabela 1). Pri tom su jednaka rastojanja od konture dela koji se preseca do najblizih tacaka na traci i rastojanja između kontura dva uzastopna izratka u uzduznom pravcu trake.

Kod primene automatskog pomeranja trake vrednosti za a i b odnosno al i bl umanjiti za 10-20% tj.

$$\begin{aligned} a &= '0.8 * a' & (1) \\ b &= '0.8 * b' & (2) \\ al &= '0.8 * al' & (3) \\ bl &= '0.8 * bl' & (4) \end{aligned}$$

strana 2++

tabela 1

Velicina dodatka a i b odnosno al i bl (Romanovski)

Debljina lima mm	Dodatak (mm)		Debljina lima mm	Dodatak (mm)	
	a i b	al i bl		a i b	al i bl
'0.3'	'1.4'	'2.3'	'4.0'	'2.5'	'3.5'
'0.5'	'1.0'	'1.8'	'5.0'	'3.0'	'4.0'
'1.0'	'1.2'	'2.0'	'6.0'	'3.5'	'4.5'
'1.5'	'1.4'	'2.2'	'7.0'	'4.0'	'5.0'
'2.0'	'1.6'	'2.5'	'8.0'	'4.5'	'5.5'
'2.5'	'1.8'	'2.8'	'9.0'	'5.0'	'6.0'
'3.0'	'2.0'	'3.0'	'10.0'	'5.5'	'6.5'
'3.5'	'2.2'	'3.2'			

$$\begin{aligned} b &= 'a' & (5) \\ bl &= 'al' & (6) \end{aligned}$$

strana 3

Prema Zubcovu dodaci između kontura dela i od ivice trake do prve konture za meke celike dati su u tabeli 2. Za druge materijale tablicne vrednosti treba pomnožiti sa koeficijentom kl i to:

- celik srednje tvrdoce i tvrdi celik.....'0.8'-'0.9'
- bronza i mesing.....'1.0'-'1.2'
- duraluminijum.....'1.0'-'1.2'
- bakar i aluminijum.....'1.2'-'1.3'
- meki nemetalni materijali  
(koza, karton i slicno).....'1.5'-'2.0'

$$\begin{aligned} a &= 'kl * a' & (7) \\ b &= 'kl * b' & (8) \\ al &= 'kl * al' & (9) \\ bl &= 'kl * bl' & (10) \end{aligned}$$

strana 4

\_tabela 2

Velicina dodatka a i b odnosno al i bl (Zubcov)

za okrugle i ovalne delove

Debljina lima (mm)	Velicina delova D (mm)									
	do 50		50-100		100-200		200-300			
	a	b	a	b	a	b	a	b		
do	'0.2'	'1.5'	'2.0'	'1.7'	'2.2'	'2.0'	'2.5'	'2.2'	'2.8'	
	@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@2.1									
	'0.2'	'0.5'	'1.2'	'1.5'	'1.4'	'1.7'	'1.6'	'1.9'	'1.8'	'2.2'
	'0.5'	'1.0'	'0.8'	'1.2'	'1.0'	'1.4'	'1.2'	'1.6'	'1.4'	'1.8'
	'1.0'	'1.5'	'1.1'	'1.5'	'1.3'	'1.7'	'1.5'	'1.9'	'1.7'	'2.1'
	'1.5'	'2.0'	'1.5'	'1.9'	'1.7'	'2.1'	'1.9'	'2.3'	'2.1'	'2.5'

strana 5

\_tabela 2 nastavak za pravougule delove

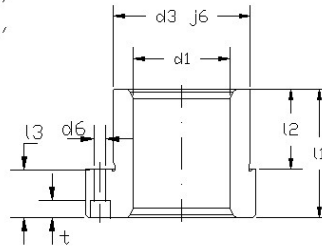
Debljina lima (mm)	Velicina delova D (mm)									
	do 50		50-100		100-200		200-300			
	al	bl	al	bl	al	bl	al	bl		
do	'0.2'	'2.0'	'2.5'	'2.5'	'3.0'	'3.0'	'3.5'	'3.5'	'4.0'	
	'0.2'	'0.5'	'1.5'	'1.8'	'1.7'	'2.0'	'2.2'	'2.5'	'2.7'	'3.0'
	@@@@@@@@@@@@@@@@@@@@^									
	'0.5'	'1.0'	'1.0'	'1.5'	'1.2'	'1.7'	'1.7'	'2.2'	'2.2'	'2.7'
	'1.0'	'1.5'	'1.4'	'1.9'	'1.6'	'2.1'	'2.1'	'2.6'	'2.6'	'3.1'
	'1.5'	'2.0'	'1.7'	'2.2'	'1.9'	'2.4'	'2.5'	'3.0'	'2.9'	'3.4'

strana 11

poglavlje lim / strana broj: 11 /

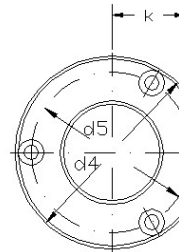
\_tabela 4

d1	d3	d4	d5	d6	l1	l2	l3	t	k
'18'	'36'	'52'	'42'	'4.5'	'36'	'22'	'14'	'4.6'	'18'
'60'	'85'	'120'	'100'	'9.0'	'80'	'50'	'30'	'9.0'	'46'



<-- |  
<-- |

's1-6'



<-- |

strana 33

\_tabela 14

d1	d3	d4	d5	d6	l1	l2	l3	t	k
'18'	'36'	'52'	'42'	'4.5'	'36'	'22'	'14'	'4.6'	'18'
'60'	'85'	'120'	'100'	'9.0'	'80'	'50'	'30'	'9.0'	'46'

strana 34

\_tabela 15

d1	d3	d4	d5	d6	l1	l2	l3	t	k
'15'	'30'	'52'	'42'	'4.5'	'50'	'32'	'18'	'4.6'	'15'
'80'	'105'	'148'	'100'	'10.0'	'80'	'50'	'30'	'9.0'	'46'

## 8. ДИСКУСИЈА О РЕЗУЛТАТИМА

Систем за пројектовање и одговарајући програмски језик који је развио аутор овог рада припада научно-технолошкој области примене рачунара за област машиноградње и припадајућих обрадних технологија.

Овај систем за пројектовање и одговарајући програмски језик у потпуности су базирани на парадигми параметарског модела система за пројектовање („parametric” *CAD* систем).

Развијени систем је функционално – у смислу параметарских модела и њихових „манипулација” – и доменски комплетан, у односу на циљ пројекта, а то је *2D* геометријско пројектовање за област, на првом месту, ка деловима од лима, плоча, платна и *2D* презентацији *3D* делова (цртање, цртежи).

Коришћење система за делове чије *2D* механичко моделирање задовољава пројектантске захтеве – делови од лима, фолија, платна – дозвољава развој нових додатних функција система, као што је функција аутоматског програмирања контроле машина алатки за обраду пројектованих делова (другим речима: екстензија развијеног система ка *CAM* системима (*CAM systems – Computer Aided Manufacturing Systems*), или развој *interface*-а за директно интегрисање система са већ постојећим системима за аутоматско програмирање контроле машина алатки (постојећим *CAM* системима). С друге стране, развијени програмски језик, такође, дозвољава интегрисање у амбијенте других система за аутоматско пројектовање – *CAD* системе, као што су *AutoCAD*, *Pro-Engineer*, *CATIA*, итд.

„Овај резултат, желим да подвучем, представља врхунски светски резултат који многе земље и истраживачки центри нису достигли, или не могу да достигну. Заправо јако мали број најразвијенијих компанија произвођача *CAD software*-а на свету нуде тржишту производе – *CAD* системе – базиране на параметарском моделирању, из простог разлога што је то данас врхунска светска технологија, за чију имплементацију је потребно комплексно мултидисциплинарно знање. Даље, формулација и имплементација једног специфичног језика за програмирање/моделирање представља други врхунски резултат, с обзиром да је пројектовање, програмирање и имплементација једног језика вероватно најтежи задатак из области рачунарских технологија.”

Цитат: 2009 Оцена Језика за *2D* параметарско геомтријско моделирање од проф.др Горана Путника, *Универсидаде до Минхо, Гумараес*, ПОРТУГАЛ

„Проматрам ли језик у подручју дефинирања геометрије, чини се да је језик замишљен за дефинирање објеката у једној равнини односно за *2D* објекте. Врло је коректно и досљедно дефиниран с модерним механизмима за параметарска и геометријска ограничења какве данас имају сви модерни *MCAD* састави (*Alibre, Inventor, SolidWorks, Solid Edge, Catia...*).”

Цитат: 2012 Оцена Језика за *2D* параметарско геомтријско моделирање од доц. др Томислава Галете, Стrojарски факултет, Славонски Брод, ХРВАТСКА

Што се тиче ограничења, систем има ограничења првенствено у односу на аспекте амбијента рада корисника (рецимо недостатак *Windows* амбијента, као и елемената иконичког интерфејса за графичко програмирање). Међутим, ови недостаци више представљају технички него квалитативни део, с обзиром на то да је најтежи део у развоју једног софтверског пакета развој тзв. *core* софтвера, а не сâм интерфејс. Аспект *3D* домена је такође важан, али он може и да се одбаци уколико се узме у обзир да је циљни домен пројектовање *2D* делова, што има високу употребну вредност у индустријама као што су метална индустрија за производе од лима, фолија, плоча и техничких платана, текстилна и кожна индустрија, дрвна индустрија итд.

Успешна примена софтверских решења код процеса пројектовања производа и технологије евидентна је у многим компанијама. Овде је значајно да компаније користе одговарајући систем са језиком или системом за геометријско моделирање. На пример, у једном релевантном испитивању корисника везаног за *CAD/CAM* системе међу највећим бројем питања било је оно које се односи управо на то да ли тај систем омогућава геометријско моделирање, које укључује опис форми са променљивим димензијама. Иако постоје многи системи, ова област није адекватно решена, јер су то системи који су често практични, али с друге стране, недостаје им добра теоријска поставка и ова решења као таква нису комплетна. У овом истраживању се покушало да се област геометријског моделирања прошири једним новим приступом код *2D* описа форми техничких објеката, чији је крајњи резултат нови језик за *2D* параметарско геометријско моделирање.

Најважније активности у оквиру овог рада дале су резултат (интересовање немачке фирме *Data M* и стављање на листу светских научних радова од стране аустралијске организације *SCIE*), а то су:

### **1. израда опште структуре језика**

Овде се подразумева да је израђена структура компјутерског програмског језика специјалне намене, али без конкретне лексичке, семантичке и синтаксичке структуре. Структура је таква да врхунски стручњаци могу на бази ове опште структуре развити разне варијанте техничког извођења језика за *2D* параметарско геометријско моделирање.

### **2. израда елемената у структури примарног (геометријског) описа ентитета и модификација**

У овој фази су урађени детаљно сви елементи у структури описа ентитета и модификација, као што су: карактеристичне тачке, координате карактеристичних тачака, карактеристични параметри, релације и релациони искази (3 типа и 2 подтипа).

### **3. лексичка, семантичка и синтаксичка структура језика за *2D* параметарско моделирање**

Израђена је детаљна лексичка, семантичка и синтаксичка структура језика за *2D* параметарско моделирање. Структура је таква да омогућава угодан рад корисника овог компјутерског програмског језика. Овде се већ даје једна варијанта техничког извођења на бази опште структуре. Ова структура, као и структура дата под тачком 1, треба да су такве да се може језик практично употребљавати коришћењем препроцесора или процесора. Дефиниције ентитета, као и завршетак израде препроцесора, омогућавају коришћење језика за *2D* параметарско геометријско моделирање у фабрикама.

У Језику за 2D параметарско геометријско моделирање развијеном у овом раду форма дата на слици 7 може се описати на много начина, а овде се дају три (ознаке ентитета су дате на слици 21).

опис 1

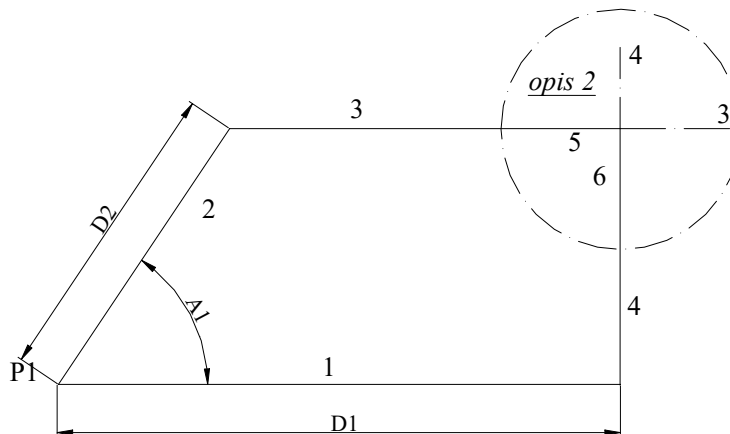
```
{1;L:[M=100,100;L=D1;A=0]}
{2;L:[M=M(1);A=(A1)%;L=D2]}
{3;L:[M=N(2);N=xn(1),yn(2)]}
{4;L:[M=N(3);N=N(1)]}
```

опис 2

```
{1;L:[M=100,100;L=D1;A=0]}
{2;L:[M=M(1);A=(A1)%;L=D2]}
{3;WL:[M=N(2);L=D1+D2;A=0]}
{4;WL:[M=N(1);L=D1+D2;a=90%]}
{5;L:[M=N(2);N=P(3,4)]}
{6;L:[M=n(1);N=P(3,4)]}
```

опис 3

```
{1;L:[M=100,100;L=D1;A=0]}
{2;L:[M=M(1);A=(A1)%;L=D2]}
{3;L:[M=N(2);/(1);L=D1-D2*cos(A1*PI/180);xn>xm]}
{4;L:[M=N(3);N=N(1)]}
```



Слика 21: Форма са слике 7. са ознакама ентитета

Очигледно да је ово једноставнији начин моделирања ове једноставне фигуре у односу на систем дат у 2. поглављу на слици 7.

У даљем тексту се даје објашњење система описа геометрије, код језика *MAPT*.

Структура реченице у овом језику је следећа:

Број команде, команда, параметри.

Овде се користе геометријски ентитети:

- тачка

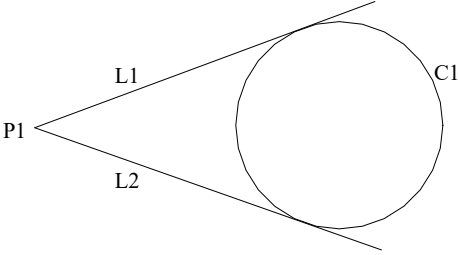
- права линија
- кружница
- профил
- крива линија
- површина

Почетним словом назива ентитета се означава о ком ентитету се ради, а уз то се даје и индекс према коме се разликују ентитети истог типа. На пример, P14 је тачка са индексом 14, L23 је права линија са индексом 23. Код дефинисања геометрије могу се користити и променљиве, као на пример  $v2r = \sin[VS] * 180$  (у овом примеру VS је претходно дефинисана променљива), при чему на првом месту у ознаци променљиве мора да буде слово V.

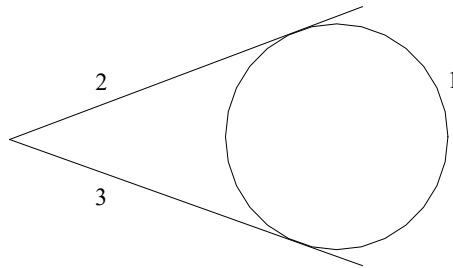
Овде ће се дати изабрани примери описа ентитета ради поређења са Језиком за 2D параметарско геометријско моделирање.

Шема је дата на левој страни, а десно општа дефиниција и примери дефинисања ентитета.

а) Права линија која пролази кроз тачку и тангира кружницу  
(МАРТ опис је у табели испод)

	<p>Општа дефиниција -----</p> <p><math>L_i = P_j, C_k, a</math> а је положај праве <math>L_i</math> у односу на кружницу <math>C_k</math>, где за вредност:</p> <p>A – тангира кружницу са горње стране, B – тангира кружницу са доње стране, L – тангира кружницу са леве стране, R – тангира кружницу са десне стране.</p> <p>Примери: L1=P1,C1,A L2=P1,C1,B L3=P2,C1,L L4=P2,C1,R</p>
--	--





Слика 22: Пример ентитета дуж тангентно на кружницу

Неки од примера дефинисања дужи за претходну шему дату на слици 21 у Језику за 2D параметарско геометријско моделирање могу да буду:

{2;L:[M=20,20;Ic(1);L=140;A<]}

{3;L:[M=20,20;Ic(1);L=b2\*cos(a3);A>]}

или

{2;L:[M=20,20;Ic(1);L=140;yn>yc(1)]}

{3;L:[M=20,20;Ic(1);L=b2\*cos(a3);yn<yc(1)]}

или

{2;L:[M=20,20;Ic(1);L=140;yn>]}

{3;L:[M=20,20;Ic(1);L=b2\*cos(a3);yn<]}

Опис дужи је овде ближи природном опису. Опис је разноврснији. Поред тога, опис је свеобухватнији; може се дати и дужина дужи. Ако се изостави дужина, подразумева се да је крајња тачка дужи на кружници. Опис је краћи код језика *MAPT*.

б) Кружница која тангира две праве линије  
(*MAPT* опис је у табели испод)

	<p>Општа дефиниција</p> <p><math>C_i = L_j, L_k, R, a, b</math></p> <p><math>a</math> и <math>b</math> је положај кружнице <math>C_i</math> у односу на праве линије <math>L_j</math> и <math>L_k</math>, слично као код примера датог под а);</p> <p><math>R</math> је радијус кружнице</p> <p>Примери дефинисања кружнице</p> <p>-----</p> <p><math>C1=L1,L2,25,B,A</math></p> <p><math>C2=L1,L2,25,L,R</math></p> <p><math>C3=L1,L2,25,A,B</math></p> <p><math>C4=L1,L2,25,R,L</math></p>
--	--

У наредном делу текста се даје опис кружнице у Језику за 2D параметарско геометријско моделирање дате на слици 22.

$$\{4;O:[IC(1);IC(2);R=25;yc>yp(1,2);xc<xp(1,2)]\}$$

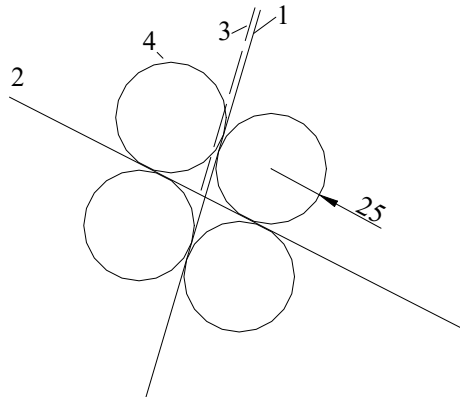
Овде су у конкретној дефиницији дата два релациона исказа, јер се издваја једна од четири кружнице.

Првим релационим исказом,  $yc>yp(1,2)$ , издвајају се две кружнице, код којих је  $y$  координата центра већа од  $y$  координате пресечне тачке дужи 1 и 2.

Другим релационим исказом,  $xc<xp(1,2)$ , издваја се једна од претходно издвојених кружница, и то она код које је  $x$  координата центра мања од  $x$  координате пресечне тачке дужи 1 и 2.

Уколико се користе фиктивни ентитети, може се ово издвајање потребне кружнице свести на један релациони исказ. То је за већину корисника погодније. Да би се описала потребна кружница са једним релационим исказом, уводи се фиктивна дуж. Почетак ове фиктивне дужи је у пресеку дужи које тангира кружница која се описује. Ова фиктивна дуж се дефинише тако да је тангирају две кружнице, а међу њима је она кружница која се описује (издваја). Тада се лако може са једним релационим исказом издвојити једна (жељена) кружница.

На пример, фиктивна дуж 3 на следећој шеми може да се дефинише као



Слика 23: Пример ентитета кружница тангентно на две дужи

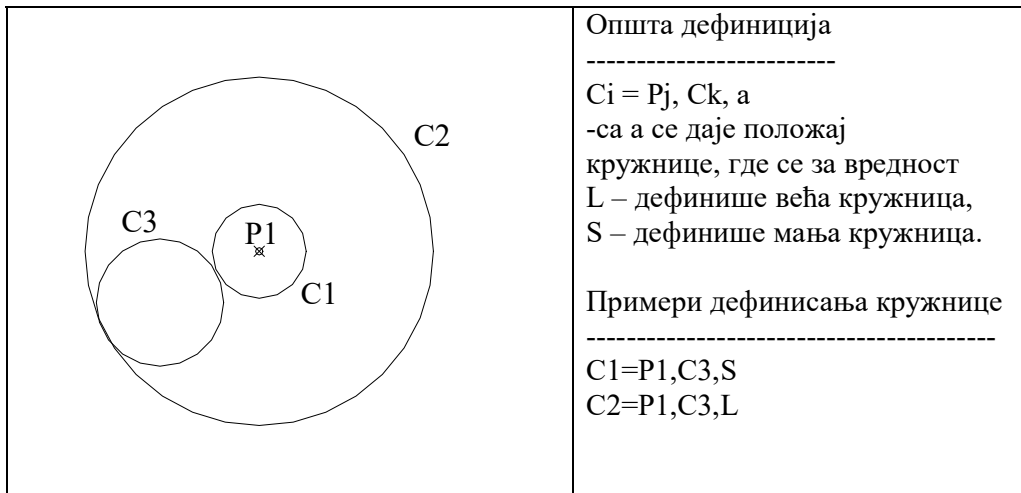
$$\{3;WL:[M=P(1,2);N=N(1)]\}$$

а жељена кружница се лако издваја од две кружнице које тангирају фиктивну дуж 3, помоћу једног релационог исказа

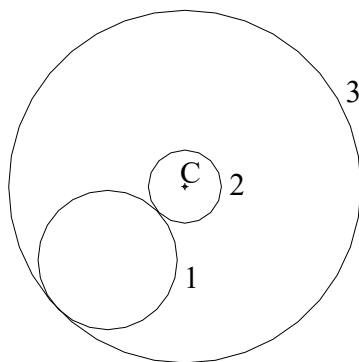
$$\{4;O:[IC(3);IC(2);R=25;yc>]\}.$$

Овде се, код Језика за 2D параметарско геометријско моделирање, природније дефинише кружница која тангира две дужи које се секу. Уз то и разноврсност описа је већа. Опис је краћи код језика *MAPT*.

в) Кружница са датом тачком центра која тангира другу кружницу  
(*MAPT* опис је у табели испод)



- Језик за параметарско геометријско моделирање



Слика 24: Пример ентитета кружница са датом тачком центра тангентно на другу кружницу

$\{2;O:[C=_;IC(1);R<]\}$   
 $\{3;O:[C=_;IC(1);R>]\}$

У претходним дефиницијама за кружницу дату на слици 23 за мању кружницу може да се употреби релациони исказ  $D<$ , односно за већу кружницу  $D>$ . Додељивање вредности за тачку C може да се дефинише на разне начине.

На ових неколико примера се може уочити разлика између ова два језика. У целини посматрано, разноврсност описа је већа код језика за параметарско геометријско моделирање, иако и у језику *MART* постоји велики број дефиниција за опис ентитета. То се види и код дефинисања сваког појединачног ентитета, а још више је то изражено када се описује нека форма. У Језику за *2D* параметарско геометријско моделирање могу се позивати све карактеристичне тачке, координате карактеристичних тачака и параметри других ентитета код описа

неког ентитета који је део неке форме. Могу се увести, ако је потребно, и фиктивни ентитети. Ово знатно олакшава опис форми.

Код Језика за  $2D$  параметарско геометријско моделирање на природнији начин се дефинишу ентитети, што његовим корисницима олакшава учење. Ово није случај код језика *MAPT*. На пример, у језику *MAPT* код тражења тачке као центра неке кружнице даје се опис  $P1=C1$ , где је  $C1$  ознака за тачку, а у другом случају код описа линије која је тангентна на кружницу даје се опис  $L1=P1,C1,A$ , где је  $C1$  ентитет на који је тангентна дуж  $L1$ . Притом је још и редослед уноса делова описа ентитета фиксан.

Опис је краћи у језику *MAPT*.

Тренутно се Језик за  $2D$  параметарско геометријско моделирање рекламира на сајту који је поставила израелска амбасада у Београду.

## 9. ДАЉА ИСТРАЖИВАЊА

У току даљих истраживања требало би се усредсредити на два главна правца. Један је да се направи професионални процесор или препроцесор за коришћење Језика за  $2D$  параметарско геометријско моделирање. Коришћење професионалног препроцесора и његову дистрибуцију на европском тржишту је имала намеру да изведе немачка фирма *Data M*. Производњу и коришћење професионалног процесора или *CAD/CAM* система је разматрала немачка фирма *LINDE*.

Стварање професионалног *CAD/CAM* система захтева већи број истраживача и велика финансијска средства. Језик за  $2D$  параметарско геометријско моделирање, као и препроцесор и процесор би морали да се прилагоде *Windows* амбијенту.

Пројектантска база података као подршка Језику за  $2D$  параметарско геометријско моделирање треба такође да буде произведена у *Windows* амбијенту, о чему је аутор овог рада излагао на два симпозијума [58][59].

Требало би извести и додатна истраживања у смислу да се формулишу нове дефиниције за моделирање ентитета које произилазе из структуре дате у овом раду. Модификација осносиметрично пресликавање је изнад исте модификације дате у другим језицима за моделирање, као и у *CAD* или *CAD/CAM* системима, јер се оса пресликавања може дефинисати на све начине на које се може уопште дефинисати дуж. И друге модификације би бар у неком делу требале да буду изнад оних које су дате у постојећим *CAD* или *CAD/CAM* системима.

За побољшање у модификацијама, на пример, овде се дају три селекције ентитета.

*Селекција помоћу изломљене линије (слика 25)[56]*

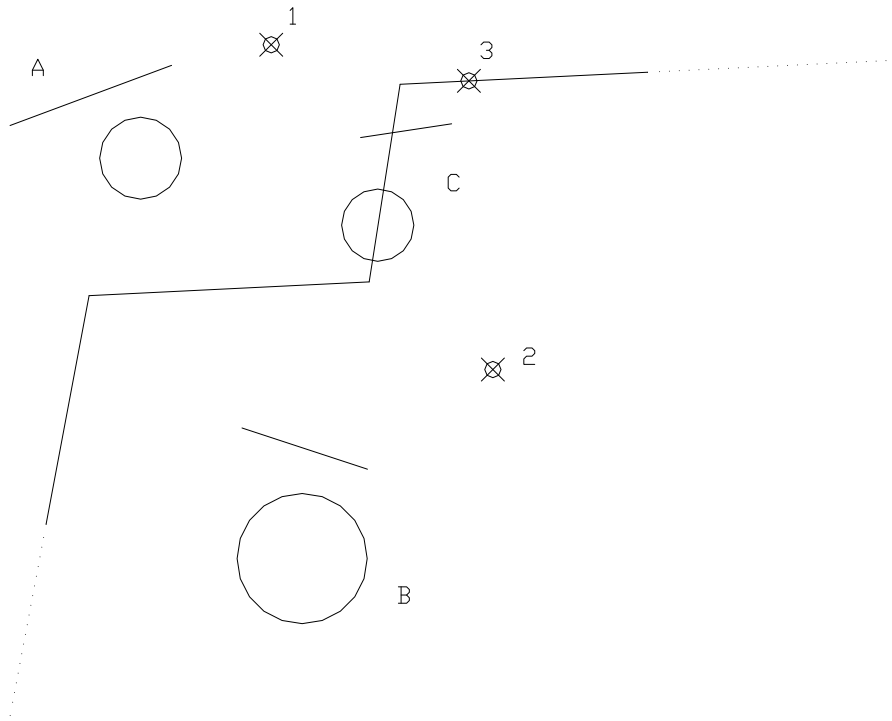
Формира се изломљена линија.

Даје се мишем страна са које се селекују ентитети или се притисне на саму изломљену линију.

Притисне се мишем на (1) на једну страну изломљене линије и селекују се ентитети дати под *A*.

Притисне се мишем на (2) на другу страну изломљене линије и селекују се ентитети дати под *B*.

Притисне се мишем (3) на изломљену линију и селекују се ентитети дати под *C* које сече ова линија.



Слика 25: Селекција помоћу изломљене линије

### *Селекција помоћу три избора (слика 26)[56]*

Бира се опција правоугаоник или многоугаоник (1).

Бира се опција Све тачке или Бар једна тачка (2).

Ако се изабере опција Све тачке, тада се селекују ентитети чије се све тачке налазе унутар или ван геометријске фигуре биране под (1).

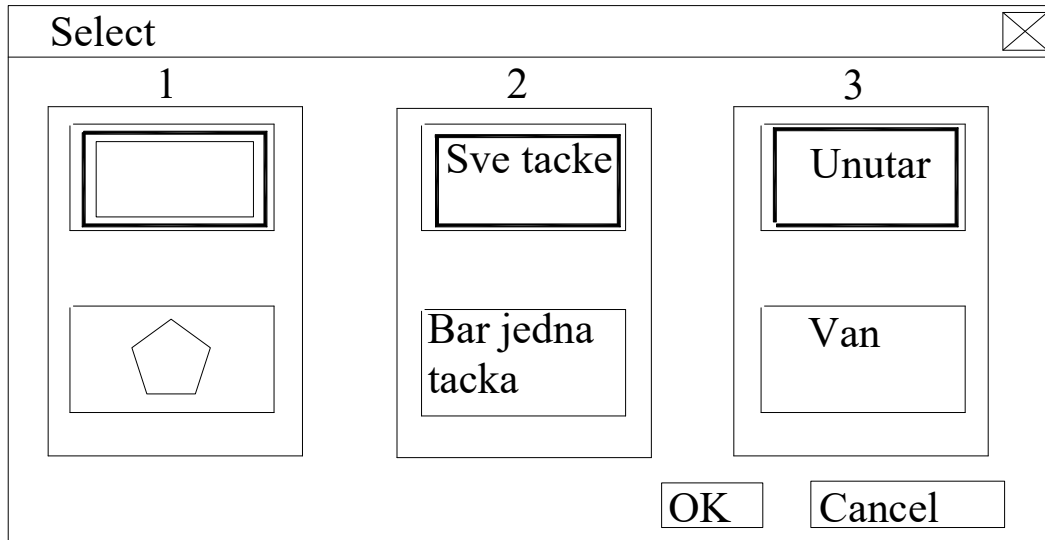
Ако се изабере опција Бар једна тачка, тада се селекују ентитети чија се бар једна тачка налази унутар или ван геометријске фигуре биране под (1).

Бира се опција Унутар или Ван.

Ако се изабере опција Унутар, тада се селекују ентитети који се налазе унутар геометријске фигуре биране избором опције дате под (1) и где се поред тога бира једна од опција дата под (2).

Ако се изабере опција Ван, тада се селекују ентитети који се налазе Ван геометријске фигуре биране избором опције дате под (1) и где се поред тога бира једна од опција дата под (2).

Формира се правоаоник или многоугаоник уносом у интерактивном раду или уносом са тастатуре две наспрамне тачке код правоугаоника или две тачке код формирања многоугаоника.



Слика 26: Селекција помоћу три избора

#### *Селекција секундарних карактеристика (метаселекција)*

Ово решење може да се користи у *CAD* систему који би био изграђен на основу решења из Језика за *2D* параметарско геометријско моделирање, а вероватно може да се користи и у другим *CAD* системима.

Select objects: *CS* – ознака да се жели избор секундарних карактеристика

of – одабере се мишем ентитет

characteristic: *LT* – ознака за тип линије (пуна линија, црта-тачка-црта, итд.),

*LC* – ознака за боју,

*LW* – ознака за дебљину линије.

Ако се притисне *Enter*, бирају се ентитети са типом линије, бојом и дебљином линије од изабраног ентитета мишем (имплицтно се подразумева да је логички везник *I*).

Може се унети само једна карактеристика, на пример *LT*, па се тако бирају сви ентитети са тим типом линије. Исто је за боју и дебљину линије.

Могу се унети и две или три карактеристике и ставити логички везник *OR* између њих, на пример *LT.or.LC* (бирају се ентитети који имају тип линије или боју изабраног ентитета).

Могу се унети и две карактеристике и ставити логички везник *AND* између њих, на пример *LT.and.LC* (бирају се ентитети који имају тип линије и боју изабраног ентитета).

Може се додати селекција према врсти ентитета:

Један приступ (мета):

Select object: *ET*

*of*

бира се мишем ентитет и бирају се сви ентитети тог типа

На пример одабере се кружница и бирају се све кружнице

*of*

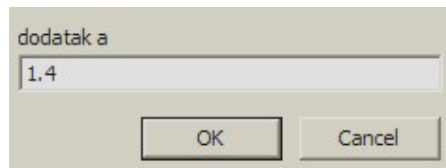
бира се мишем други ентитет или *Enter* за прекид избора.

Овде ће се само приказати како би изгледао *Windows* амбијент за пројектантску базу података [58,59].

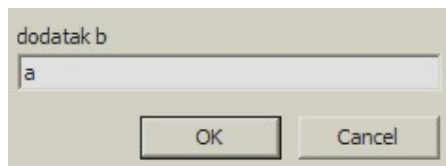
Пребацивање са једне на другу страну у књизи (пројектантској бази података) врши се уобичајено помоћу траке за претраживање.

Када се доведе показивач на константу и притисне комбинација тастера тастатура-миш, појављује се оквир за дијалог, као што дато на слици 27, где може да се прихвати или измени вредност изабрана из табеле.

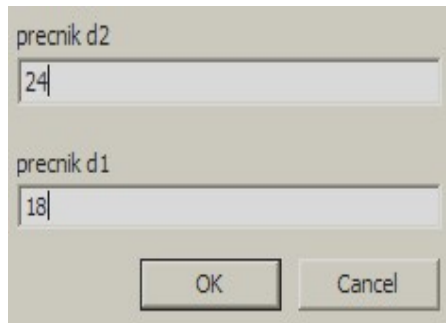
Када се доведе показивач на вредност додатка датог у облику формуле б='а' и притисне комбинација тастера тастатура-миш, појављује се оквир за дијалог, као што је дато на слици 28, где може да се прихвати или измени изабрана формула.



Слика 27: Избор константе



Слика 28: Избор формуле



Слика 29: Избор параметара из табеле

Ако у једном реду у табели има више вредности, а врши се избор свих вредности у том реду, притисне се друга комбинација тастера тастатура-миш, када се показивач доведе испред прве вредности у том реду. Отвара се оквир за дијалог као на слици 29, где су вредности изабраних параметара поређане једна испод друге. Овде се вредности параметара које су издвојене из табеле могу прихватити или изменити.

Прва страна је она страна на којој се налази садржај. Довођењем показивача на број стране или на назив појединог поглавља и притиском на леви тастер миша прелази се на страну на којој почиње то поглавље.

Изглед једног дела базе података потребне код пројектовања за слике на којима се налази графичка представа једног машинског елемента је сличан изгледу датог у поглављу Пројектантска база податка као подршка Језику за 2D параметарско геометријско моделирање. Бира се слика притиском миша на идентификацију слике, а пре тога су одабрани параметри потребни за цртање тог машинског елемента на пре описан начин. Ако подаци који стоје у опису слике нису одабрани или добијени прорачуном, појављује се оквир за дијалог као на слици 28 [58,59], али без унесених података, и очекује се унос података у празна поља за параметре који стоје у опису неког машинског елемента у Језику за 2D параметарско геометријско моделирање развијеног у овом раду.



## 10. ЗАКЉУЧНЕ НАПОМЕНЕ

Главна предност Језика за  $2D$  параметарско геометријско моделирање лежи пре свега у расположивости свих могућих варијанти описа ентитета.

У примарном (геометријском) опису ентитета могу се користити комбинације од елемената, као што су карактеристичне тачке, координате карактеристичних тачака и други геометријски параметри, као што су дужина, ширина, висина, угао, пречник, полупречник, дистанца између карактеристичних тачака и слично, релациони искази за положај једног ентитета у односу на други стварни или фиктивни ентитет са релацијама паралелно, управно, тангентно и слично, као и релациони искази за издвајање једног од два или више могућа скупа тачака који произилазе из осталог/претходног дела примарног (геометријског) описа. Такође се може користити у примарном делу описа и опис функцијама (домен и кодомен). У свим изразима код примарног описа за опис тачака, односно координата тачака или било којих других параметара, као и код релационих исказа за издвајање једног од два или више могућа скупа тачака и код дефинисања израза за домен и кодомен када се ентитет описује функцијама, поред константи, променљивих, аритметичких операција, степеновања и стандардних математичких функција, те заграда, могу се користити координате и други геометријски параметри, а и цели изрази код описа ентитета функцијама, као и координате тачака пресека ентитета. Оваква структура овог израза омогућује да се могу дефинисати форме са конкретним димензијама, општим димензијама и форме са комбинацијом општих и конкретних димензија. Уопште узев, структура примарног описа ентитета је таква да омогућује велики избор могућих варијанти за опис неке форме.

Дозвољена је употреба и фиктивних ентитета, којим се олакшава опис неких других ентитета.

Све карактеристике стварних или фиктивних ентитета могу да се користе код описа других ентитета. Укратко, опис ентитета дат у Језику за  $2D$  параметарско геометријско моделирање је погодан за корисника зато што је опис кратак, а уз то велика је разноврсност описа, и осим тога, разумљивост описа за корисника је веома велика.

Рад садржи кратке дефиниције за моделирање линијских сегмената паралелних  $x$  и  $y$  оси. Ове дефиниције су погодне за интерактиван рад код цртања и моделирања.

Опис ентитета дат у Језику за  $2D$  параметарско геометријско моделирање је погодан за корисника јер се ентитети могу описати у већем броју варијанти описа, а програмски модул у којем је садржан механизам за анализу описа и синтезу програма омогућује такав излаз да се он може реализовати у крајњем резултату у пакету *AutoCAD*.

У првим истраживачким поглављима *Општа структура* и *Елементи у структури примарног (геометријског) описа ентитета* дата је структура Језика за  $2D$  параметарско геометријско моделирање без конкретне лексичке, семантичке и синтаксичке структуре. На основу овог поглавља могу се развити језици за  $2D$  параметарско геометријско моделирање са различитим верзијама лексичке,

семантичке и синтатичке структуре или геометријски део код *CAD* система односно *SAM* система.

У истраживачком поглављу *Дефиниције примарног (геометријског) описа ентитета* дато је мноштво дефиниција за оријентисану дуж да се покаже да из претходног истраживања произилазе скоро све дефиниције за дуж које уопште постоје.

У следећем истраживачком поглављу *Структура језика за 2D параметарско геометријско моделирање* дата је конкретна поставка лексичке, семантичке и синтаксичке структуре Језика за 2D параметарско геометријско моделирање. Затим је дата једна скраћена верзија лексичке, семантичке и синтаксичке структуре Језика за 2D параметарско геометријско моделирање.

У поглављу *Дефиниције команди (командних линија) за моделирање ентитета* дате су, уз графички приказ, потпуно применљиве опште дефиниције ентитета, као и примери дефинисања ентитета. Овде су дате дефиниције за:

- оријентисану дуж
- кружницу
- кружни лук
- криву линију
- правоугаоник
- многоугаоник.

Показано је како се моделирају криве линије у параметарском и непараметарском облику.

Овде су дате и неке специфичне дефиниције са релационим исказима.

У поглављу *Примена језика за 2D параметарско геометријско моделирање* дато је коришћење фиктивних ентитета код 2D параметарског геометријског моделирања.

Поред тога, овде су дати и примери описа форми машинских елемената, и то они програми за опис дате форме који садрже и релације и релационе исказе, као и фиктивне ентитете код једног примера.

Модификација пресликавања није овде дата пошто је детаљно објашњена у претходном поглављу.

Овде је дата и програмска реализација кривих линија, односно како препроцесор производи мале *LISP* програме за реализацију једне криве линије у *AutoCAD*-у.

Затим је у овом поглављу дат кратак опис језика за 2D параметарско геометријско моделирање и препроцесора за његово коришћење да се види како је могуће програмски реализовати коришћење Језика за 2D параметарско геометријско моделирање.

Даље, развијена је пројектантска база података као подршка Језику за 2D параметарско геометријско моделирање.

Пројектант код пројектовања машинских конструкција користи разне књиге и приручнике, итд. Осим тога, он укључује и своје писане или неписане личне искуствене информације. Стога се овде од тога пошло – да се све ово интегрише, односно представи и користи у што природнијем облику. Развијен је ради тога један део система који служи као помоћ пројектанту, а у њему су интегрисане све врсте информација. Дакле, све врсте информација (константе, формуле, дијаграми и разни други графови, текстови обавештења и објашњења) из разних књига,

приручника, каталога, као и информације из искуства појединог пројектанта или групе пројектаната потребне за пројектовање спајају се у једну целину. Као такве се и користе. Пројектант овде, уместо разних информација датих на уобичајен начин у књигама, користи пројектантску базу података, коју знатно лакше листа и притом директније користи у раду. Једино обавештења и објашњења се не користе директно, већ на исти начин као из уобичајених извора информација код класичног пројектовања.

Дакле, уместо уноса параметара неког машинског елемента када се реализује програм којим је моделиран неки машински елемент, могуће је исти добити и прорачуном у пројектантској бази података.

Препроцесор за коришћење Језика за  $2D$  параметарско геометријско моделирање је уједно и програм за коришћење пројектантске базе података. Анализа описа ентитета је први део рада препроцесора. После тога следи производња низа малих програма којим се добијају конкретне вредности за цртање појединих ентитета. Овде је, у ствари, примењено метапрограмирање. Један велики програм омогућује производњу низа малих програма који се извршавају.

## 11. ЛИТЕРАТУРА

1. Albers A., Leon-Rovira H., Aguayo H., Maier T., Development of an enginecrankshaft in a frame-work of computer-aided innovation. *Computers in Industry*, 2009, 60(8): 604–12.
2. Allerkamp D., Bottcher G., Wolter F.-E., Brady A.C., Qu J., Summers I. R., A Vibrotactile Approach to Tactile Rendering, *The Visual Computer*, 2007, 23(2):97–108.
3. Amenta H., Peters T.J., Russell A.C., Computational topology: ambient isotopic approximation of 2-manifolds, *Theoretical Computer Science*, Volume 305, Issues 1–3, 2003, 3–15.
4. Aziz EI-Sayed, Chassapis C., Knowledge-based geometry generation for spur and helical gears. *Concurrent Engineering: Research and Applications*, 2002, 10(3): 251–61.
5. Bottcher G., *Medial Axis and Haptics*. Master's Thesis, Leibniz Universitat Hannover, 2004.
6. Baohua, S., Mingxi, T., Frazer, J. H. and Haicheng, Y., Stroke-based Intelligent Sketching Interface, *Proceeding of the 5th Asia Pacific Conference on Computer Human Interaction APCHI 2002*, 500–509.
7. Bettig B., Shah J.J., Derivation of a standard set of geometric constraints for parametric modeling and data exchange, *Comput Aided Des* 2001, 33:17–33.
8. Biswas A., Shapiro V. and Tsukanov I., *Heterogeneous Material Modeling with Distance Fields*. Technical Report, University of Wisconsin-Madison, Mechanical Engineering Department, Spatial Automation Laboratory, SAL 2002-4.
9. Bremer P.T., Hamann B., Kreylos O., Wolter F.-E., Simplification of Closed Triangulated Surfaces Using Simulated Annealing, *Proceedings of the Fifth International Conference on Mathematical Methods for Curves and Surfaces*, Oslo, 2000.
10. Budynas R., Hisbitt K., Shigley J.E., *Mechanical Engineering Design*, McGraw-Hill, United States of America, 2001.
11. Cao B-W, Chen J-J, Huang Z-G, Zheng Y., CAD/CAE integration framework with layered software architecture, In: *Proc. of 11th IEEE international conference on computer-aided design and computer graphics*, Huangshan (China), 2009, 410–5.
12. Chase G.H., Cheney DC, Bailey TT, Johnson D., Identifying, correcting and avoiding errors in computer aided design models which affect interoperability, *J Comput Inform Sci Eng*, 2001, 1: 156–66.
13. Chen G, Ma Y-S, Thimm G, Tang S-H., Unified feature based integration of design and process planning, In: *Hinduja Srichand, editor, Proceedings of the international MATADOR conference*, Springer, 2004, 63–8.
14. Chen G, Ma Y-S, Thimm G, Tang S-H., Unified feature modeling scheme for the integration of CAD and CAE, *Computer-Aided Design and Applications*, 2004, 1(1-4): 595–601.
15. Choi G-H, Mun D, Han S., Exchange of CAD part models based on the macro-parametric approach. *Int J CAD/CAM*, 2002, 2: 13–21.
16. Colombo G., Mosca A., Sartori F., Towards the design of intelligent CAD systems: an ontological approach, *Advanced Engineering Informatics*, 2007, 21(2): 153–68.
17. Contero, M., Haya, F., Jorge, J. and Conesa, J.: CIGRO: A Minimal Instruction Set Calligraphic Interface for Sketch-Based Modeling, *Lecture Notes in Computer Science*, 2669, 2003, 549–558.
18. Cukovic S., Devedzic G., Ghionea I., Automatic determination of grinding tool profile for helical surfaces machining using CATIA/VB interface, *Scientific Bulletin of the University Politehnica of Bucharest, Series D*, vol.72, issue 2/2010, Bucharest, 2010.
19. Cvetkovic D., *CAD/CAM Teorija i praksa upravljanja proizvodnjom*, Univerzitet Singidunum, Beograd, 2010.
20. Dassault Systèmes – SolidWorks Corporation, SolidWorks 2006, <http://www.solidworks.com/sw/products/mechanical-engineering-cad-software.htm>, приступ: 22.6.2009.
21. D-Cubed, Cambridge, UK, <http://www.d-cubed.co.uk/>
22. Delcam plc – ArtCAM. <http://www.artcam.com/>, приступ: 18.6.2009.
23. Deng Y-M, Britton GA, Lam YC, Tor SB, Ma Y-S., A feature-based CAD-CAE integration model for injection molded product design, *International Journal of Production Research*, 2002, 40(15): 3737–50.
24. Edelsbrunner H., Harer J. and Zomorodian A., Hierarchical Morse Complexes for Piecewise Linear 2-Manifolds, *Symposium on Computational Geometry*, 2001.

25. Fayek A.R., Sun Z., Afuzzy expert systemfor design performance predictionand evaluation, Canadian Journal of Civil Engineering 28, 2001, 1–25.
26. Foucault G, Cuilliere J-C, Francois V, Leon J-C, Maranzana R., Adaptation of CAD model topology for finite element analysis, Computer-Aided Design, 2008, 40(2): 176–96.
27. Fredy J., About Solid Modeling by using feature-based models, <http://www.cs.uml.edu/~fjara/oldcourses/review3.html>, [приступ 4.8.2009](#).
28. Fuh, J. Y. H. and Li, W. D., Advances in collaborative CAD: the-state-of-the art, Computer-Aided Design, 2005, 37(5): 571–581.
29. Gerardo Salas Bolanõs, Stephen Mann, Sanjeev Bedi, Targeted WebCAD, Computer Aided Design and Applications, Vol 6, No. 5, 2009, 639–644.
30. Ghionea I., Stude of computer aided design and simulation importance in the conception anddevelopment phases of a product, Proceedings of the IMC 2013 International Multidisciplin-ary Conference, Horth University of Baia Mare România – University College of Hyiregyhaza, Bessenyei Publishing House, Hungary, 2013, 55–58.
31. Gujarathi GP, Ma Y-S., Generative CAD and CAE integration using common datamodel, In: Proc of the 6th annual IEEE conference on automation science andengineering, Toronto (Canada), 2010.
32. Horváth R., Kátai L., Balázs M., Molnár L., CAD book, University of Technology and Economics, Faculty of Mechanical Engineering, Szent Istvan Egyetem, 2012.
33. <http://docslide.net/documents/projektovanjepomocuracunara-2008-09.html>, [януар 2017](#)
34. Jianjiang Zeng, Wenliang Chen, Quiling Ding, A Web-based CAD system, Journal of Materials Processing Technology, Vol 139, 2003, 229–232.
35. Johansson J., Manufacturability analysis using integrated kbe, CAD and FEM, In: Proceedings of the ASME international design engineering technicalconferences and computers and information in engineering conference, DETC 2008, vol. 5, 2009, 191–200.
36. Jovanović M., Projektovanje računarom, predavanja, 2010.
37. Junhvan K., Standardized data exchange of CAD models with design intent, Computer-Aided Design 40, 2008, 760–777.
38. Kao Yung-Chou, Cheng Hsin-Yu, She Chen-Hua., Development of an integrated CAD/CAE/CAM system on taper-tipped thread-rolling die-plates, Journal of Materials Processing Technology, 2006, 177(1-3): 98–103.
39. Khemlani L., Autodesk Revit: Implementation in Practice, Arcwiz, Fremont CA, 2004, 19.
40. Klein, M., Sayama, H., Faratin, P., Bar-Yam, Y., The Dynamics of Collaborative Design: Insights from Complex Systems and Hegotiation Research, Concurrent Engineering, 11, 2003, 201–209.
41. Lee G., Eastman C.M., Sacks R., Eliciting information for product modeling using process modeling, Data & Knowledge Engineering, 62, 2007, 292–307.
42. Lee SH., A CAD-CAE integration approach using feature-based multiresolution and multi-abstraction modeling techniques, Computer-Aided Design, 2005, 37(9): 941–55.
43. Liu Q, Liu W, Ruan F, Qiu H., Parameters' automated optimization in sheet metalforming process, Journal of Materials Processing Technology, 2007, 187–188: 159–63.
44. Long, A.C., Landay, J.A., Rowe, L.A., Michiels, J., Visual Similarity of Pen Gestures, Proceedings of Human Factors in Computer Systems (SIGCHI), 2000, 360–367.
45. MacKrell J., Exchanging product design data: Business benefits of the collaboration gateway. CIMdata, <http://www.CIMdata.com>; [приступ: април 2004](#).
46. Monedero J., Parametric design. A review and some experiences, <http://info.tuwien.ac.at/ecaade/proc/moneder/moneder.htm>, [приступ: 27.7.2009](#).
47. Monedero J., Parametric design: a review and some experiences, Automation in Construction, 2000, 9(4): 369–77.
48. Mun D, Han S, Kim J, Oh Y., A set of standard modeling commandsfor the history-based parametric approach, Comput Aided Des 2003, 35: 1171–9.
49. Papalambros P.Y., The optimization paradigm in engineering design: promisesand challenges, Computer-Aided Design 2002, 34:939–51.
50. Park HS, Pham HH., Design of conformal cooling channels for an automotivepart, International Journal of Automotive Technology, 2009, 10: 87–93.
51. Patil L, Dutta D, Sriram R., Ontology-based exchange of product datasemantics, IEEE J Automation Sci Eng, 2005, 2: 213–25.

52. Patrikalakis HM, Maekawa T., Shape interrogation for computer aided design and manufacturing, New York: Springer, 2002.
53. Peak RS., Characterizing fine-grained associativity gaps: a preliminary study of CAD-CAE model interoperability, In: Proc. of the ASME design engineering technical conference, vol. 1A, 2003, 573–80.
54. Penoyer J.A., Burnett G., Fawcett D.J., Liou S-Y., Knowledge based product life cycle systems: principles of integration of KBE and C3P, Computer-Aided Design 2000, 32: 311–20.
55. Pereira, J., Jorge, J., Branco, V., Hunes, F., Towards calligraphic interfaces: sketching 3D scenes with gestures and context icons, WSCG'2000, Conference Proc. Skala V, 2000.
56. Popović Ž., Arandelović I., Nikolić J., A Language for 2D Parametric and Geometric Modelling – PrESMod and Entity Selection, Contemporary Engineering Sciences, Bugarska, Vol. 6, No. 5, 2013, 203–211, World Scientific Index (WSI)  
<http://www.m-hikari.com/ces/ces2013/ces5-8-2013/3316.html>
57. Popović Ž., Arandelović I., Jezik za geometrijsko modeliranje PrESMod - neke specifične definicije, I Medunarodno savetovanje *Informatika u proizvodnom i poslovnom menadžmentu* - IPOM, Doboj, 2004.
58. Popović Ž., Arandelović I., Projektantska baza podataka kao podrška jeziku za modeliranje PrESMod, Zbornik radova 29. JUPITER konferencije, Beograd, 2005.
59. Popović Ž., Dimitrijević P., Vučićević M., Podrška projektu korišćenjem baze podataka i jezika za modeliranje PrESMod, 9. internacionalni simpozijum Project management JUMPA, Zlatibor, 2007.
60. Popović Ž., Postupak geometrijskog modeliranja (2D) pomoću kompjutera korišćenjem relacionih iskaza, 28. JUPITER konferencija, Beograd, 2002.
61. Popović Ž., Presmod modeling language, Intertribo 2009, X International Symposium, Vysoke Tatry, Slovak Republik.
62. Popović Ž., Putnik G. D., Arandjelović I. (2008) PrESmod Language for 2D Geometric Modelling- Some Specific Definitions, in: Cagil G., Kubat C., Oztemel E, Taskin H (Eds.) Intelligent & Manufacturing Systems – Features, Strategies and Innovation (Proceedings of the 6th International Conference on Intelligent & Manufacturing Systems, Sakarya, Turkey, 2008.
63. Popović Ž., Some specific procedures of geometrical modeling in the presmod language for 2D parameter geometric modeling, Tehnicki vjesnik / Technical Gazette, Vol. 24 / No. 2, 2017, 413-417, ISSN 1330-3651 (Print), DOI: 10.17559/TV-20160621114351, SCI rad.
64. Popović Ž., Neke specifične definicije kod jezika za modeliranje, PrESMod, 33. međunarodni kongres o automatizaciji – HIPHEM, Beograd, 2012.
65. Popović Ž., Arandelović I., The PrESMod Language for Modeling and an Approach to Entity Selection, XVII International Conference on MATERIAL FLOW, MACHINES AND DEVICES IN INDUSTRY, University of Belgrade, Faculty of Mechanical Engineering, 2004.
66. Popović, Ž., The PrESMod Language for Modeling (2D), in Proceedings of the 7th International Scientific and Expert Conference of the International TEAM Society, Belgrade, 2015.
67. Pratt MJ, Anderson WD, Ranger T., Towards the standardized exchange of parameterized feature-based CAD models. Comput Aided Des 2005, 37: 1251–65.
68. Radhakrishnan P, Subramanyan S, Raju V., CAD/CAM/CIM. Daryaganj (Delhi, IHD): Hew Age International, 2008, 123–245.
69. Rao M., CAD / CAM, Department of Mechanical Engineering, Indian Institute of Technology, Delhi, 2010.
70. Rappoport A., An architecture for universal CAD data exchange, In: Proc. 2003 ACM solid modeling symposium. New York: ACM Press, 2003, 266–269.
71. Resh WF., Computer aided engineering in engine design, International Journal of Vehicle Design, 2000, 23(1): 30–7.
72. Roy R, Hinduja S, Teti R., Recent advances in engineering design optimisation: challenges and future trends, CIRP Annals-Manufacturing Technology, 2008, 57: 697–715.
73. Sacks R., Evaluation of the economic impact of computer-integration in precast concrete construction, ASCE Journal of Computing in Civil Engineering, 18, 2004, 301–312.
74. Salomon D., Curves and Surfaces for Computer Graphics, Horthridge, U.S.A. Springer, 2006.

75. Sang H. L., A CAD-CAE integration approach using feature-based multi-resolution and multi-abstraction modelling techniques, *Computer-Aided Design*, 37, 2005, 941–955.
76. Schmidt, K., Wagner, I., Coordinative Practices and Artifacts in Architectural Design and Planning, *Computer Supported Cooperative Work*, 13, 2004, 349–408.
77. Schreier P., Modeling: CAD/CAE: closing the CAD/CAE gap, *Scientific ComputingWorld*, No. 109, 2009/2010, 4.
78. Schweikardt, E., Gross, M.D., Digital Clay: deriving digital models from freehand sketches, *Automation in Construction*, 9, 2000, 107–115.
89. Seo T-S, Lee Y, Cheon S-U, Han S, Patil L, Dutta D., Sharing CAD models based on feature ontology of commands history. *Int J CAD/CAM*, 2005, 5.
80. Shephard MS, Beall MW, O'Bara RM, Webster BE., Toward simulation-based design, *Finite Elements in Analysis and Design*, 2004, 40(12): 1575–98.
81. Smit MS, Bronsvort WF., Integration of design and analysis models, *Computer-Aided Design and Applications*, 2009, 6(6): 795–808.
82. Song W, Keane A, Rees R, Bhaskar A, Bagnall S., Turbine blade fir-treeroot design optimization using intelligent CAD and finite element analysis, *Computer and Structure*, 2002, 80: 1853–67.
83. Spitz S, Rappoport A., Integrated feature-based and geometric CAD data exchange, In: *Proc. 2004 ACM solid modeling and applications symposium*, New York: ACM Press, 2004.
84. Stewart M., Arnold K., *Surface production operations volume 1, gas-liquid and liquid-liquid separators*, Gulf Professional Publishing, 2009.
85. Stiteler M., *Construction History And Parametric S: Improving afford ability through intelligent CAD data exchange, CHAPS program final report*, Advanced Technology Institute, 5300 International Boulevard, HorthCharleston, SC 29418, USA, 2004.
86. Sturts C.S., Griffis F.H.B., Pricing engineering services, *Journal of Management in Engineering* 21, 200, 56–62.
87. Sun Wei, Ma Qinyi, Chen Shuang, A framework for automated finite element analysis with an ontology-based approach, *Journal of Mechanical Science and Technology*, 2010, 23(12): 3209–20.
88. Sung H. Ahn, V. Sundararajan, Charles Smith, Balaji Kannan, Roshan D Souza, Ganping Sun, Ashish Mahole, Paul K. Wright, JaeHo Kim, Sara McMains, Jordan Smith, Carlo H. Séquin., *CyberCut: An Internet based CAD-CAM system*, ASME, Vol 1, 2001, 52–59.
89. Turner, A., Chapmann, D., and Penn, A., *Sketching space*, *Computers & Graphics*, 24, 2000, 869–879.
90. Uputstvo za jezik AutoLISP
91. Uputstvo za jezik MAPT
92. Uputstvo za sistem ProEHGIHEER
93. Van der Velden Dr Alex., CAD to CAE process automation through iSIGHT-FD, *Proceedings of the ASME Turbo Expo, 2007*, 1: 87–93.
94. Walsh K, J.G. Wacker B., *Engineering Productivity Measurements II*, Construction Industry Institute, Austin, Texas, 2004, 272.
95. Wang GG, Shan S., Review of metamodeling techniques in support of engineering design optimization, *Journal of Mechanical Design*, 2007, 129(2): 370–80.
96. Wang Y., *Constraint-enabled design information representation formechnical products over the internet (Ph.D. thesis)*, University of Pittsburgh, 2003.
97. Wang, C.C.L. and Yuen M.M.F., *Freeform Extrusion by Sketched Input*, *Computers & Graphics*, 27, 2003, 255–263 .
98. WEISBERG D., *THE ENGINEERING DESIGN REVOLUTION, ON-LINE BOOK*, 2008.
99. Werner DC, Karl-G F., *Engineering in reverse-a holistic extension of CAD*, In: *Papers of the workshop, Hew Trends in Engineering Design*, Balatonfured, Hungary, 2003, 91–4.
100. Wikipedia, *Computer Aided Design*, [http://en.wikipedia.org/wiki/Computer-aided\\_design](http://en.wikipedia.org/wiki/Computer-aided_design), приступ: 21.6.2009.
101. Xu Bin, Chen Han., *An integrated method of CAD, CAE and multi-objective optimization*, In: *Proceedings of the 2009 IEEE 10th international conference on computer-aided industrial design & conceptual design, CAID&CD 2009*, 2009, 1010–4.

102. Xu Xiangyang, Weiss Ulrich, Gao Guoan, The integration of CAD/CAM/CAE based on multi model technology in the development of cylinder head, *International Journal of Automotive Technology*, 2002, 3(2): 47–52.
- 103.. Xu XY, Wang YY., Multi-model technology and its application in the integration of CAD/CAM/CAE, *Journal of Materials Processing Technology*, 2002, 129 (1–3): 563–7.
104. Yip DCY, Law MC, Cheng KP, Lau KH, Barnes S., Knowledge intensive CAD in product design validation, *International Journal of Knowledge-Based and Intelligent Engineering Systems*, 2005, 9(2): 45–61.
105. Zeng S, Kim EJ, Mocko GM, Xiao A, Peak R, Mistree F., Systematic design method for information modeling in CAD/CAE, In: *Proceedings of the ASME design engineering technical conference*, vol. 1 B, 2003, 657–68.
106. Zeng S, Peak RS, Xiao A, Sitaraman S., A knowledge-based FEA modeling method for highly coupled variable topology multi-body problems, *Engineering with Computers*, 2008, 24(4): 359–81.
107. Zhengfeng Yan, Zhengfeng Jiang., Research on the integrated design of dual mass fly wheel CAD/CAE/CAM, *Key Engineering Materials*, 2009, 407–408: 169–73.



## 12. ПРИЛОЗИ

### Прилог 1: Писмо немачке фирме *data M*



data M Software GmbH  
Am Marschallfeld 17  
D-83626 Valley/Oberlindern  
Germany  
Tel.: (+49) 08024 640-0  
Fax: (+49) 08024 640-300  
E-Mail: [datam@datam.de](mailto:datam@datam.de)  
Internet: <http://www.datam.de>



Valley, 03.01.08

#### Informationsmaterial

Sehr geehrter Herr Popovic,

vielen Dank für Ihr Schreiben vom Oktober 2007. Als Anlage erhalten Sie schon mal Informationsmaterial über unser Softwarelösungen COPRA MetalBender und COPRA Rollform.

Wir sind an Ihrer Lösung für die 2D-Modellierung interessiert. Ist es möglich, detailliertere Informationen darüber zu bekommen und evtl. eine Vorführung?

Es wäre außerdem gut, wenn wir Ihre Telefon / Faxnummer hätten und, falls vorhanden, Ihre E-Mailadresse. Unsere Adresse hat sich übrigens vor einiger Zeit geändert. Sie ist jetzt:

data M Software GmbH  
Am Marschallfeld 17  
83626 Valley/Oberlindern  
Tel.: 08024/640-0  
Fax: 08024/640-300

Wir würden uns freuen, bald von Ihnen zu hören und verbleiben

mit freundlichen Grüßen

data M Software GmbH

  
Claudia Kury

Sitz der Gesellschaft: Valley/Oberlindern  
Registriergericht München HRB-Nr.: 81828  
Ust-ID-Nr.: DE 129022754

Geschäftsführer:  
Dipl. Ing. Albert Sedwimer  
Dipl. Ing. Stefan Freitag  
Dipl. Ing., Dipl. Wirtsch.-Ing. Andreas Bachthaler

Deutsche Bank AG München  
Kto.-Nr.: 501 610 0 - BLZ: 700 700 10

**Прилог 2: Одлука аустралијске организације SCIE да стави рад из ове докторске дисертације на своју листу светских научних радова**

24. 2. 2015. Australian Society For Commerce Industry & Engineering wsi-2013-chapter-21

Email: [contact@scie.org.au](mailto:contact@scie.org.au)

Home Conferences Proceedings Books Journals Learning Centre World Scientist Index News FAQ

Back to SCIE Journals

**OPEN ACCESS**

**World Scientist Index (WSI) Reports**

Each year, SCIE will produce a << World Scientist Index (WSI) Report >> to summarise research achievement for worldwide scientists in collaboration with wikiscientist.

Wikiscientist is a FREE platform funded by wiki program. The wikiscientist platform enables worldwide scientists to:

1. Build online personal academic biography
2. Share research interests, research findings
3. Record academic publication history
4. Demonstrate academic publication list
5. Find and connect with researchers, academics, scientists in the same fields in a worldwide scope.
6. To be invited as a reviewer in peer review papers
7. To be invited in as chair in international conferences

**Wikiscientist**

[Click to be indexed in Wikiscientist program](#)

SCIE Australia is in a strategic partnership with the wikiscientist in producing the << World Scientist Index Reports >>, which is fully funded by Wiki Program. Scientists that has been indexed in wikiscientist is also entitled to be a member of SCIE Australia.

**<< World Scientists Index (WSI) 2013 Reports >>**

Chapter 21:

**Cite this report as:**  
 Australian Society for Commerce Industry Engineering (SCIE), << World Scientist Index 2013 Reports >>, Sydney Australia, 2013. Available online <http://www.scie.org.au/world-scientist-index>

**Disclaimer:** SCIE Australia summaries world wide latest research findings. This is a free index service for open access journals. The index does not contain contents of the papers; the copy-right belongs to the author(s). The WSI service is only for paper indexing purpose. The WSI follows the Open Access Policy 3.0, which enables SCIE to distribute and share the materials freely. Should you have any questions or wish to make any changes in the index, please contact: [wsi@scie.org.au](mailto:wsi@scie.org.au)

Article#	Article Title & Authors
10001	<b>Airborne observations of trace gases over boreal Canada during BORTAS: campaign climatology, airmass analysis and enhancement ratios</b> S. J. O'Shea, G. Allen, M. R. Gallagher, S. J. S. Baugwitz, S. M. Mangayath, M. Le Brunn, J. B. A. Hubler, C. J. Percival, A. T. Archibald, D. E. Olson, M. Pinnigstein, P. I. Palmer, A. C. Lewis
10002	<b>Airborne Particulate Matter and Meteorological Interactions during the Haze Period in Malaysia</b> Carolye Perias, Norawi Abdulah, and Norita Sabirin
10003	<b>Airborne Particulate Matter and Meteorological Interactions during the Haze Period in Malaysia</b> Carolye Perias, Norawi Abdulah, Norita Sabirin
10004	<b>Airborne sun photometer PLASMA: concept, measurements, comparison of aerosol extinction vertical profile with lidar</b> F. Karol, D. Tardif, P. Gobush, C. Versaende, J. F. Malin, L. Blain, T. Pridon, A. Morin, A. Chaboulay
10005	<b>Airborne Thermal Data Identifies Groundwater Discharge at the North-Western Coast of the Dead Sea</b> Ulrich Müller (Department Geomathematik-Neurobiologie, Helmholtz-Centre for Environmental Research, D-06120 Halle, Germany), Friedhelm Schwandt (Sub-Department Geobotanik Assessment and Resource Sensing, Federal Institute for Geosciences and Natural Resources (BGR), D-10655 Hannover, Germany), Richard Glasziou (Remote Sensing Group, Helmholtz Institute Freiberg of Resource Technology, D-09599 Freiberg, Germany), Stefan Geyer (Department Catchment Hydrology, Helmholtz-Centre for Environmental Research, D-06120 Halle, Germany), Martin Sauer (Department Applied Geology, Geoscience Centre, University of Göttingen, D-37077 Göttingen, Germany), Christian Gebert (Department Carbonate Hydrology, Helmholtz-Centre for Environmental Research, D-06120 Halle, Germany)
10006	<b>Alnched chirally-coupled-core Yb-fiber femtosecond oscillator with &gt;10W average power</b> Ma Xinyuan, Gaharavadas Arumugam, Zhu Chang, Chang Guangqiang, Chen Jiang-Hua, Kijneer-Franz V
10007	<b>AIR COOLING IN AUTOMOBILES USING VORTEX TUBE REFRIGERATION SYSTEM</b> B.SREENIVASA KUMAR REDDY, Prof. A.GOVINDARAJULU

- 10297 **A Numerical Investigation of Biological Mucus on Gas Transport in a Cough Machine Simulating Trachea: Effect of Surfactant as a Sol Phase Layer**  
Dipak Kumar Sanyal, Aditya Pal, Rany
- 10298 **A laminar flow unit for the care of critically ill newborn infants**  
Perez JM, Golombek SG, Fajardo C, Soler A
- 10299 **A Landesman-Lazer condition for the boundary-value problem  $-u'' + u^p = h(u)$  with periodic boundary conditions**  
Quinn A, Morris, Stephen B, Robinson
- 10300 **A landmark paper in HIV research?**  
Paula Zaitava-Borron
- 10301 **Ni and Si Influences on Hydrogen Embrittlement of Carbide-Free Bainitic Steel**  
Yongjun Li (State Key Laboratory of Metastable Materials Science and Technology, Tsinghua University, Qianxuesun Road 100084, China), Cheng Chen (State Key Laboratory of Metastable Materials Science and Technology, Tsinghua University, Qianxuesun Road 100084, China), Fackeng Zhang (State Key Laboratory of Metastable Materials Science and Technology, Tsinghua University, Qianxuesun Road 100084, China)
- 10302 **A language for 2D parametric and geometric modelling - PRISM and entity selection**  
Zeljko Popovic, Ivan Brundalovic, Jelena Nikolic
- 10303 **A Language for Fuzzy Statistical Database**  
C.P. Kott, S.Gugiani
- 10304 **Alantolactone Induces Apoptosis in HepG2 Cells through GSH Depletion, Inhibition of STAT3 Activation, and Mitochondrial Dysfunction**  
Muhammad Khan, Ting Li, Muhammad Khalid Ahmad Khan, Akbar Rasool, Faisal Nawaz, Mehsun Sun, Tongxun Zhang, Tanghui Mo
- 10305 **Alan Turing and the origins of complexity**  
Marta Delgado, Miguel Angel (Departamento de Física Teórica I, Universidad Complutense, España)
- 10306 **Alan Turing and the origins of modern Gaussian elimination**  
Dapico, Práxim M. (Instituto de Ciencias Matemáticas CSIC-UAM-UC3M-UCM and Departamento de Matemáticas, Universidad Carlos III de Madrid, España)
- 10307 **Alan Turing: person of the XXth century?**  
Sánchez-Roa, José M. (Universidad Autónoma de Madrid - Real Academia Española, España)
- 10308 **Alan Turing's work on morphogenesis**  
Horens, Miguel A. (Departamento de Matemática Aplicada, Facultad de Matemáticas, Universidad Complutense de Madrid, España)
- 10309 **A Laparoscopic Single-Site Surgery Approach to Mesh Sacrospinorraphy**  
Ferdin Belvaen-Ribbaen, Anirudh Gang, Marc J. N. C. Karim
- 10310 **A Laplace type problem for an irregular lattice and**  
G. Corini, M. Straka
- 10311 **A Laplace type problem for an irregular trapezium lattice and body test rectangle**  
A. Puglisi, G. Corini, D. Barilla
- 10312 **A Laplace type problem for regular lattices with convex-concave cell and obstacles rhombus**  
G. Corini, A. Puglisi, E. Sema
- 10313 **A Laplace type problem for two hexagonal lattices of Delone with obstacles**  
D. Barilla, G. Corini, Marco Straka, A. Puglisi
- 10314 **Alarcón, R. (1986). Psicología, pobreza y subdesarrollo. Lima: Instituto Nacional de Investigación y Desarrollo de la Educación, 153 pgs.**  
Ramos León (Pontificia Universidad Católica del Perú)
- 10315 **Alarcón, R. D. (1990). Identidad de la psiquiatría latinoamericana. Voces y expresiones en torno a una ciencia solidaria. México, D.F.: Siglo Veintiuno Editores, 670 p.**  
Ramos León (Instituto de Salud Mental Delgado-Vigochi)
- 10316 **Alarcón, Reynaldo. (1994) El pensamiento psicológico de Walter Blumenthal. Lima: CONCYTEC, 220 p.**  
Mora Vargas Lopez
- 10317 **À LA RECHERCHE DE L'ORIGINE: L'Écriture du souvenir dans le salon de Wurttemberg de Pascal Quignard et le projet proustien**  
Clément Foukitcher
- 10318 **A large and functionally diverse family of Fhd2 genes in safflower (Carthamus tinctorius L.)**  
Cao Shuang, Zhu Jie-Rong, Wood Craig C, Green Alan G, Singh Savinder P, Liu Lina, Liu Qing
- 10319 **A Large Celiac Artery Aneurysm**  
Harinder Singh Boffi, Kawal Negi
- 10320 **A Large Celiac Artery Aneurysm**  
Harinder Singh Boffi, Kawal Negi
- 10321 **A Large Cohort Study of Hypothyroidism and Hyperthyroidism in Relation to Gynecologic Cancers**  
Shelley S. Twigg, Erik Alexander, Bernard Barnes, Gary Carhan, Inmaculada De Vito, Jae H. Kang, Angela S. Kieck, Richard Stevens
- 10322 **A large rotation of the polarization angle of the TeV Blazar W Comae**  
Borner, Erik, Silvio Marco, Brian David
- 10323 **A large-scale, high-resolution hydrological model parameter dataset for climate change impact assessment for the conterminous United States**  
A. A. Oshfeldt, S.-C. Kim, M. Arifoglu, B. Naz, G. Tootle
- 10324 **A large-scale neural network model of the influence of neuromodulatory levels on working memory and behavior**  
Michael Avery (University of California, Irvine), Jeffrey L. Dworkin (University of California, Irvine)
- 10325 **A Large-Scale Preparation Method of High Purity C-Phycocyanin**  
Suying Wang, Guizhen Zhao, Wenjun Song
- 10326 **A large-scale RF-based Indoor Localization System Using Low-complexity Gaussian filter and improved Bayesian inference**  
L. Xiao, Y. Yin, X.N. Wu, J. W. Wang
- 10327 **A large semi-synthetic single-chain Fv phage display library based on chicken immunoglobulin genes**  
Jonathan Frances, Frédéric Jousset, Mathieu Corvellec, Malak Terzian, van Huyenquien Hwaier, Milica Duhrenia, de Pleuro Dion H
- 10328 **A Large Span Crossbeam Vibration Frequencies Analysis Based on an Analogous Beam Method**  
Zhiqing Liu, Bing Luo, Tongsheng Zhao, Wenming Jiang, Lipeng Cai
- 10329 **A large supernumerary bone at the bregma and ectopism co-occurring in the skull of an ancient roman in Serbia**  
Radović P. (National Museum Belgrade, Belgrade), Mladović-Budžević Nasta (Institute of Archaeology, Belgrade)
- 10330 **Alarming Presentation of a Seemingly Common Condition in a Patient with Renal Impairment**  
Mohammed Badreddin, Mohammed Fawzi Khalil, Jena Al Saleh, Rawan Simpa
- 10331 **Alaska Dental Health Aide Program**  
Sarah Shogren-Coxe, Mary Willard
- 10332 **Alan, Poor Yorick: Digging Up the Dead to Make Medical Diagnosis**  
Khalidoff Maria, Jefferson Richard, Harmon Jessica, Harmon Jessica, Anandjyoti Resnik, Mostafaei Farhad
- 10333 **"Alan poor Yorick": What retrospective analysis of cranial skulls can tell us about the impact of environmental factors on health**  
Luciano O. Valenzuela, Renee Chase, Lawrence McGill, Sharon Miller, Mark Nelson, Karl G. Lutz

### Прилог 3: Део препроцесора за коришћење Језика за 2D параметарско геометријско моделирање

```
(DEFUN C:ULAZ()
(setq prosWW "lim")
  (if(/= nil (findfile (strcat "" prosWW ".dwg"))))
  (progn
(command "insert" (strcat "" prosWW) "0,0" "" "" "")
  )(progn

      (if(/= nil (findfile (strcat "b:" prosWW)))
      (progn
      (command "insert" (strcat "b:" prosWW) "0,0" "" "" "")
      )(progn

      (if(/= nil (findfile (strcat "a:" prosWW ".dwg")))
      (progn
      (command "insert" (strcat "a:" prosWW) "0,0" "" "" "")
      )(progn))

      ))

  ))
(command "layer" "off" "")
(command)
(command)
)

(defun C:RAD()
(command "layer" "off" "")
(command)
(command)

(command "limits" "0,0" "800,600")
(command)

(command "zoom" "w" "0,0" "16,12")
(command)

(SETQ RIS1WW "")(SETQ RIS2WW "")(SETQ RIS3WW "")(SETQ HRISWW 0)
(setq limWW nil)
  (if(/= nil (findfile "d:funkc.tas"))
  (progn
  (setq fff (open "funkc.tas" "r"))
  (setq tasfWW (read-line fff))
  (close fff))(progn (setq tasfWW "@")))

  (setq kredfWW 1)
  (setq tbuWW 1)(setq xpWW 0.1)(setq ypWW 11.5)
```

```

(setq wWW prosWW)
(setq wiWW (strcat prosWW ".izb"))
(setq fiz (open wiWW "w"))
(setq liWW "neizlaz")

(setq kWW 0)
(while(AHD(/= liWW "izlaz")(/= liWW "IZLAZ"))

(setq qWW (itoa kWW))
(setq wqWW (strcat wWW qWW))
(setq strWW (atoi qWW))
(setq bsWW qWW)
(setq rbsWW (atoi bsWW))

                                (if(/= nil (findfile (strcat "" wqWW)))
                                (progn
(command "layer" "on" wqWW "")(princ) ^P
(STAMPA)
                                )(progn(setq nestrWW (strcat "nema strane " wqWW))
                                (print nestrWW)
                                ( princ) ^P
                                ))

(setq liWW (getstring))

(SETQ LIWW (STRCASE LIWW))

(if(= (substr liww 1 3) "AG")(progn
(command "layer" "off" wqWW "")(princ) ^P
(setq qqrceo liww)(setq qqkoci 1)(setq qqm1 "")(setq qqm2 ""))(MAR)
(if(= (itoa (atoi QQizmed)) QQizmed)(progn

(setq TGww (strcat prosww qqizmed))
(command "layer" "on" TGww "")(graphscr)(princ) ^P
                                )(progn))
)(progn))

                                (redli)
                                (if(AHD(= jrWW 1)(= HESTRWW HIL))
                                (progn(lired)
                                (progn(anal1)))
(SETQ HESTRWW HIL)

)

(izlazi)

(COMMAHD "LAYER" "OFF" WQWW "")(COMMAHD "LAYER" "OFF" "" "" "")
)

(defun izlazi()
                                (close fiz)

                                (if(/= nil (findfile (strcat "" wiWW)))
                                (progn
(setq fiz (open wiWW "r"))
(ZIZB)(setq nrisWW 0)

```

```

                (close fiz)
(setq fiz (open wiWW "w"))(close fiz)
                )(progn))
)

(defun stampa()
(princ "")(terpri)
(setq gst (open wqww "r"))
(setq unan_ 0)
(while (/= nil (setq zg_ (read-line gst)))
(princ zg_)(TERPRI)
(SETQ unAH_ (+ unAH_ 1))
)
(close gst)
(if(< unan_ 22)(progn

(SETQ HAH_ 1)
(WHILE(< HAH_ (- 25 unan_))
(terpri)
(setq nan_ (+ nan_ 1))
)
)(progn))

(setq nan_ nil)
(TEXTSCR)
)

(defun anal1()
(setq plusWW 0)
(setq niWW (strlen liWW))
(setq n1WW 1)
(while(<= n1WW niWW)
(setq QQzg (substr liWW n1WW 1))
(if(= QQzg "+")
(progn(setq plusWW (+ plusWW 1)))
(progn))
(setq n1WW (+ n1WW 1))
)
)

(if(> plusWW 0)
(progn(command "layer" "off" wqWW "")(graphscr)(princ) ^P
(setq kWW (+ kWW plusWW))

                (if(/= TGww nil)(progn
                (COMMAHD "LAYER" "OFF" "*" "" "" "")(graphscr)(princ) ^P
                (setq TGww nil) )(progn))

)(progn(anal2 )))
)

(defun anal2()
(setq minusWW 0)
(setq niWW (strlen liWW))
(setq n1WW 1)
(while(<= n1WW niWW)
(setq QQzg (substr liWW n1WW 1))

```

```

(if(= QQzg "-")
(progn(setq minusWW (+ minusWW 1)))
(progn)
(setq niWW (+ niWW 1))
)

(if(> minusWW 0)
(progn(command "layer" "off" wqWW "")(graphscr)(princ) ^P
(setq kWW (- kWW minusWW))

                                (if(/= TGww nil)(progn
(COMMAHD "LAYER" "OFF" "*" "" "")(graphscr)(princ) ^P
(setq TGww nil) )(progn))

)(progn)
)

(defun redli()

(setq jrWW 0)
(setq niWW (strlen liWW))
(setq niWW 1)
(while(<= niWW niWW)
(setq QQzg (substr liWW niWW 1))
(if(or(= QQzg "r")(= QQzg "R"))
(progn(setq jrWW (+ jrWW 1)))
(progn

(if(= QQzg "?")
(progn

                                (close fiz)
                                (if(/= nil (findfile (strcat "" wiWW)))
                                (progn(setq fiz (open wiWW "r"))
                                (ZIZB)(setq nrisWW 0)
                                (close fiz)
                                )(progn))

                                (setq fiz (open wiWW "w"))

                                (if(= (getstring) "REG")(progn (Command "regen" ))(progn))

)(progn))

))
(setq niWW (+ niWW 1))
)
)

(defun lired()

(setq iredWW 0)
(setq jsWW 0)

                                (while(/= jsWW 1)

```

```

(setq liWW (getstring)) (SETQ LIWW (STRCASE LIWW))
(setq krecWW nil)(setq krec1WW nil)(setq krec2WW nil)
(if(or(= liWW "I")(= liWW "IZBOR"))
  (progn (textscr) (textizb))
  (progn))
(if(or(= liWW "GI")(= liWW "GRAF"))
  (progn
    (graphscr)(Command "Regen")(princ "izbor misem-{Izlaz >Enter misem -drugo dugme}")
    (SETQ GLBDWW LBDWW)(SETQ GWQWW WQWW) (GRAFIZB))
  (progn))
                                (SADR)
(sTRAHA)
(analrp)
(setq iredWW (+ iredWW PlusWW))
(setq iredWW (- iredWW minusWW))

                                (setq lbdWW wqWW)(setq lbd (open wqWW "r"))
(if(>= iredWW 21)(progn(setq iredWW 21))
  (progn ))
(if(<= iredWW 0)(progn(setq iredWW 1))(progn))
                                (setq nredWW 1)
                                (SETQ IREDWW (+ IREDWW 1))
(while(<= nredWW iredWW)
  (setq redWW (read-line lbd))
  (if(= iredWW nredWW)
    (progn(setq reddWW redWW)
      (setq numWW (- iredWW 1))
      (princ reddWW)
      )
    (progn))
    (setq nredWW (+ nredWW 1))
  )
  (close lbd)
)
)

(defun sTRAHA()
  (setq jsWW 0)
  (setq nliWW (strlen liWW))

  (setq n1WW 1)
  (while(<= n1WW nliWW)
    (setq QQzg (substr liWW n1WW 1))
    (if(or(= QQZG "s")(= QQZG "S"))
      (progn(setq jsWW (+ jsWW 1)))
      (progn
        ))
    (setq n1WW (+ n1WW 1))
  )
)
)

```



```

(defun analrp()
  (setq plusWW 0)
  (setq niWW (strlen liWW))
  (setq n1WW 1)
  (while(<= n1WW niWW)
    (setq QQZG (substr liWW n1WW 1))
    (if(or(= QQZG "+")(= QQZG " "))
      (progn(setq plusWW (+ plusWW 1)))
      (progn))
    (setq n1WW (+ n1WW 1))
  )
  (if(and(> n1WW 1)(= plusww 1))(progn (setq qqrceo (strcat liww "#"))
    (setq qqm1 "+" )(setq qqm2 "#")(setq qqkoci 1) (MAR)
    (if(/= (atoi qqizmed) "0")(progn(setq plusww (atoi qqizmed)) )(progn))
  )(progn
  (analm)
  ))
)

```

```

(defun analrm()
  (setq minusWW 0)
  (setq niWW (strlen liWW))
  (setq n1WW 1)
  (while(<= n1WW niWW)
    (setq QQZG (substr liWW n1WW 1))
    (if(= QQZG "-")
      (progn(setq minusWW (+ minusWW 1)))
      (progn))
    (setq n1WW (+ n1WW 1))
  )
)

```

```

;sada lista strane prema sadrzaju dobro
(DEFUH SADR()
  (setq brostrWW "")
  (if(OR(= LIWW "SA")(= liWW "SADRZAJ"))(progn
  (command "layer" "off" wqWW "")(graphscr)(princ) ^P
  (setq kWW 0)(setq liWW "s"))(progn))

```

```

(IF(= LIWW "<")
  (PROGH
  (setq niWW (strlen reddWW))

```

```

(SETQ HITWW HLIWW)

```

```

(SETQ BIH1WW (SUBSTR REDDWW HITWW 1))
(SETQ BIHuWW "")

```

```

(WHILE (AHD(> HITWW 1)(/= BIH1WW "."))

```

```

(SETQ BIHuWW (STRCAT BIH1WW BIHuWW))
(SETQ HITWW (- HITWW 1))

```

```

(SETQ BIH1WW (SUBSTR REDDWW HITWW 1))

```

```

)

```

```
(IF(= "." (SUBSTR REDDWW (- HITWW 1) 1))
(PROGH
```

```
(SETQ KWW (ATOI BIHUWW))(SETQ LIWW "s")
(SETQ BIH1WW HIL)
(SETQ BIHUWW HIL)
(SETQ HITWW HIL)
)(PROGH))
)(PROGH))
)
```

```
(defun textizb()
(textscr)
(setq rediWW (getstring)) (if(= rediWW "")(progn(praz10))(progn))
```

```
(if(and(/= rediWW "")(/= reddWW nil))
(progn
  (setq rik1WW (substr rediWW 1 1))
  (IF(and(= rik1WW "^") (= rediWW "^"))
  (PROGH (izborc))
  (PROGH(izbord)))
)
(progn))
)
```

```
(defun praz10()
(terpri)
(setq r10WW "")(setq r10praWW " ")(setq rpraWW "")
(setq nr10WW 0)
(while(and(= rediWW "")(<= nr10WW 6))
(princ reddWW)(terpri)
(setq r10WW (strcat r10WW "@@@@@@@@@@"))(setq rpraWW (strcat rpraWW r10praWW))
(princ rpraWW)(setq nr10WW (+ nr10WW 1))
(setq rediWW (getstring)) (terpri)
)
(setq rediWW (strcat r10WW rediWW))
)
```

```
(DEFUH IZBORC())
```

```
(SETQ HAWW (STRLEH REDDWW))
(SETQ iiWW 1)
(WHILE(< iiWW HAWW)
(ODBAC)
(VADIR)
(setq nvrWW (+ numWW (* rbsWW 20)))
(SETQ HVRCWW (ITOA HVRWW)) (if(/= nil ikolWW)(progn (SETQ IKOLCWW (ITOA IKOLWW))
(if(/= krecWW "")
(progn
```

```
(KRECFSL)
)(progn))
```

```
)(progn))
```

```
)  
)
```

```
(DEFUN ODBAC()  
  (setq amWW (substr reddWW iiWW 1))  
  (while (and(/= amWW "")(<= iiWW naWW))  
    (setq iiWW (+ iiWW 1))  
    (setq amWW (substr reddWW iiWW 1))  
  )  
)
```

```
(defun vadir()  
  (setq krecWW "")  
  (setq iiWW (+ iiWW 1))  
  (setq amWW (substr reddWW iiWW 1))  
  (setq xirWW 0)  
  (while (and(/= amWW "")(<= iiWW naWW))  
    (setq xirWW (+ xirWW 1))  
    (if(= xirWW 1)  
      (progn(setq ikolWW (- iiWW 1))  
              (progn))  
    )  
  )  
  (setq iiWW (+ iiWW 1))  
  (setq krecWW (strcat krecWW amWW))  
  (setq amWW (substr reddWW iiWW 1))  
  )  
  (setq iiWW (+ iiWW 1))  
  )
```

```
(defun izbord()  
  (setq naWW (strlen reddWW))  
  (setq nbiWW (strlen rediWW))  
  
  (if(> nbiWW naWW)(progn(setq nabWW nbiWW))(progn(setq nabWW naWW)))  
    (setq iWW 1)  
    (while(<= iWW nabWW)  
      (setq JPIKWW 0)  
  
      (odbi)  
      (odbvad)  
      (setq nvrWW (+ numWW (* rbsWW 20)))  
  
      (SETQ HVRCWW (ITOA HVRWW))  
        (if(/= nil IKOLWW)  
          (progn (SETQ IKOLCWW (ITOA IKOLWW))  
                  (if(and(/= krec2WW "")(<= jpikWW 0))  
                    (progn  
  
                      (setq krecWW krec2WW)(KRECfSL)  
  
                      )  
                    (progn ))  
                    (if(> jpikWW 0) (progn  
  
                      (setq krecWW krec1WW)(KRECfSL)  
                      (setq jpikWW 0)  
                      )(progn))  
                    )  
                    (progn))
```

```

)
)

(defun kLub()
(setq qqm1 "")(setq qqm2 "")(setq qqrceo krecWW)(setq qqkoji 1)
(MAR)
(if(= nil qqizmed)(= "" qqizmed))
(progn(setq opc_ KRECWW)
)(progn(setq opc_ QQIZMED)
(setq qqm1 "")(setq qqm2 "")(setq qqrceo krecWW)(setq qqkoji 1)
(MAR)
(SETQ KRECWW QQIZMED)
))
)

(defun KRECfsl()
(if(<= ikolWW 1)(progn (setq zjWW " "))(progn
(setq zjWW (substr reddWW (- ikolWW 1) 1))
))
(if(<= ikolWW 2)(progn )(progn
(if(= (substr reddWW (- ikolWW 2) 1) ".")
(progn(setq zjww ":=")
)(progn))
))
)

(KRECfsl1)
)

(defun KRECfSL1()
(setq nrisWW (+ nrisWW 1))
(setq zslWW (substr reddWW (+ ikolWW 1) 3))

(setq zslWW (strcase zslWW))
(if(= nrisWW 10)(= zjWW "=")(= zslWW "SL-")(= zjww ":="))
(progn

(close fiz)
(if(/= nil (findfile (strcat "" wiWW)))
(progn(setq fiz (open wiWW "r"))
(ZIZB)(setq nrisWW 0)
(close fiz)
)(progn))

(setq fiz (open wiWW "w"))

)(progn))

(IF(= ZSLWW "SL-")(PROGn
(KLUB)

```

```

(jepored)(DRLOC )(PROGH))

(IF(= zjww ":=")(PROGH
(BEPORED) )(PROGH

(if(= zjWW "=")(progn (JEPORED)

(setq isacuWW iWW)
(setq fm_ krecWW)(FRM)(setq krecWW p0WWW)
(setq iWW isacuWW)(setq isacuWW nil)

(prin1 krecWW fiz)(write-line "" fiz)
)(progn

(write-line krecWW fiz)
))

(write-line nvrCWW fiz)(write-line ikolCWW fiz)

))
(SETQ KRECWW HIL)(SETQ KREC1WW HIL)(SETQ KREC2WW HIL)
(setq zjww nil)(SETQ ZSLWW HIL)(setq fm nil)
)

(defun odbi()
(setq iaposWW 0)(setq daposWW 0)
(setq am1WW (substr reddWW iWW 1))
(apos)
(while(and(= iaposWW 0)(<= iWW nbiWW))
(setq iWW (+ iWW 1))
(setq am1WW (substr reddWW iWW 1))
(apos)
)
(if(= iaposWW 1)
(progn(setq ikoloWW iWW))(progn))
)

(defun odbvad()
(setq krec1WW "")
(setq krec2WW "")
(setq jpikWW 0)
(setq am1WW (substr reddWW iWW 1))
(setq am2WW (substr rediWW iWW 1))
(while(and(> iaposWW 0)(<= iWW nabWW))
(if(= am2WW "^")
(progn (setq jpikWW (+ jpikWW 1)))
(progn))
)

(setq iWW (+ iWW 1))
(setq am1WW (substr reddWW iWW 1))
(setq am2WW (substr rediWW iWW 1))
(setq krec1WW (strcat krec1WW am1WW))

```

```

(if(and(/= am2WW tasfWW)(/= am2WW "@")(/= am2WW "^"))
  (progn(setq krec2WW (strcat krec2WW am2WW)))
  (progn))

(if(= am1WW "")
  (progn(setq iaposWW (- iaposWW 1)) (setq daposWW (+ daposWW 1)) )
  (progn))
)
(if(and(>= jpikWW 1)(= iaposWW 1)(= daposWW 1) )(progn
  (zadrec)      (progn))
  (setq iWW (+ iWW 1))

(if(> jpikWW 0)(progn(setq ikolWW ikoloWW)
  (setq nkrec1WW (strlen krec1WW))(setq manje1WW (- nkrec1WW 1))
  (setq krec1WW (substr krec1WW 1 manje1WW))
  )(progn))
  (if(and (= jpikWW 0)(or(/= krec2WW "")(= iaposWW 1)))
    (progn(setq ikolWW ikoloWW)(duzarec)
  (setq nkrec1WW (strlen krec1WW))(setq manje1WW (- nkrec1WW 1))
  (setq krec1WW (substr krec1WW 1 manje1WW))
  )      (progn))

(setq iaposWW 0)
)

(defun duzarec()

  (setq ldWW iww)
  (setq krec2dWW "")
  (setq amdWW (substr rediWW ldWW 1))
  (while(and(/= amdWW tasfWW)(/= amdWW "@")(<= ldWW nbiWW))
    (if(and(/= amdWW "^")(/= amdWW "@"))(progn
  (setq krec2dWW (strcat krec2dWW amdWW))      )(progn))
  (setq ldWW (+ ldWW 1))
  (setq amdWW (substr rediWW ldWW 1))

  )
  (if(/= krec2dWW "")(progn (setq iWW (+ nabWW 1))
  (setq krec2WW (strcat krec2WW krec2dWW)))
  (progn))
  )

(defun zadrec()

  (while(and(/= am1WW "")(> iaposWW 0))
  (setq iWW (+ iWW 1))
  (setq am1WW (substr reddWW i 1))
  (setq krec1WW (strcat krec1WW am1WW))
  (if(= am1WW "") (progn(setq iaposWW 0))(progn))
  )
  (setq iWW (+ nabWW 1))
  )

(defun apos()
  (if(= am1WW "")
  (progn(setq iaposWW (+ iaposWW 1)) (setq daposWW (+ daposWW 1)) )
  )

```

```

(progn)
)

(defun grafizb()
  (setq redWW "")(setq reddWW "")
  (graphscr)
  (setq xytabWW "0.5,0.5")
                                     (while(/= xytabWW nil)
  (setq xytabWW (getpoint))
    (if(= nil xytabWW)(progn(setq x1WW 2.)(setq y1WW 0.))(progn
  (setq x1WW (car xytabWW))
  (setq y1WW (car (cdr xytabWW)))
    ))
  (if(and(> x1WW 0.1)(< x1WW 16.))(and(> y1WW 1.5)(< y1WW 11.5)))
  (progn
    (setq kolWW 0)
    (while(>= x1WW 0.1)
      (setq X1WW (- X1WW 0.2))
      (setq kolWW (+ kolWW 1))
    )
    (setq nvredWW 0)
    (while(<= y1WW 11.5)
      (setq y1WW (+ y1WW 0.5))
      (setq nvredWW (+ nvredWW 1))
    )
    (pronarec)
  )
  (progn
    (if(and(or(> x1WW 16.1)(< x1WW 0.1))(or(< y1WW 11.5)(> y1WW 1.5)))
      (progn
        (setq nvredWW 0)
        (while(<= y1WW 11.5)
          (setq y1WW (+ y1WW 0.5))
          (setq nvredWW (+ nvredWW 1))
        )
        (pronred)
      )
    )
  )
  (progn)
))

)

(defun pronarec()
  (setq Glbd (open GwqWW "r"))
  (if(>= nvredWW 21)(progn(setq nvredWW 21))
  (progn ))
  (if(<= nvredWW 0)(progn(setq nvredWW 1))(progn)
    (setq nredWW 1)
    (SETQ nnREDWW (+ nvREDWW 1))
    (WHILE(and(<= nredWW nnredWW)(/= reddWW nil))
      (if(= nil (setq redWW (read-line Glbd)))
        (progn(prompt"\nnema tog reda-kraj teksta na zadnjoj strani")
          (setq nredWW nnredWW)(setq redWW nil))(progn))
    )
    (if(= nredWW nnredWW)

```

```

        (progn(setq reddWW redWW)
          (setq numWW nvredWW)(terpri)(princ reddWW) )
      (progn)
      (setq nredWW (+ nredWW 1))
    )
  (CLOSE GLBD)
  (setq redi1WW
  "@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
  @@"
  (SETQ REDI2WW "@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@")
  (SETQ REDIWW (STRCAT REDI1WW REDI2WW))
  (SETQ PKOLWW (- KOLWW 1))
  (SETQ REDIP (SUBSTR REDIWW 1 PKOLWW))
  (SETQ REDIWW (STRCAT REDIP "^"))

  (textscr) (if(/= nil reddWW)(progn(IZBORD))(progn(setq reddWW "")))
)

(defun pronred()
  (setq Gldb (open GwqWW "r"))
  (if(>= nvredWW 21)(progn(setq nvredWW 21))
  (progn ))
  (if(<= nvredWW 0)(progn(setq nvredWW 1))(progn))
    (setq nredWW 1)
    (SETQ nnREDWW (+ nvREDWW 1))
    (WHILE(and(<= nredWW nnredWW)(/= reddWW nil))
      (if(= nil (setq redWW (read-line Gldb)))
        (progn(prompt"\nnema tog reda-kraj teksta na zadnjoj strani")
        (setq nredWW nnredWW) (setq redWW nil))(progn))

      (if(= nredWW nnredWW)
        (progn(setq reddWW redWW)
          (setq numWW nvredWW)(terpri)(princ reddWW)(terpri))
        (progn))
        (setq nredWW (+ nredWW 1))
      )
    (CLOSE GLBD)
    (if(/= nil reddWW)(progn(izborc))(progn(setq reddWW "")))
)

```

```

(defun JEPORED()
  (setq jepWW 1)
  (SETQ XIS1WW (list "" krecWW))
  (SETQ XIS2WW (list "" nvrWW))
  (SETQ XIS3WW (list "" ikolcWW))
  (PORED)
)

```

```

(if(= RIWW RWW)(progn(setq krecWW RSIFWW)
  (progn))
)

```

```

(defun BEPORED()
  (setq jepWW 1)
  (SETQ XIS1WW (list "" krecWW))
)

```



```
(SETQ XIS2WW (list "" nvrCWW))
(SETQ XIS3WW (list "" iKolcWW))
(setq prosWW "lim")
```

```
(setq jepww 1)
(setq datsiWW (strcat "sifr." prosWW))
```

```
(UPORED)
```

```
)
```

```
-----
;-----dovde formula-----
```

```
(defun ZIZB()
```

```
(setq ris1WW "") (setq ris2WW "") (setq ris3WW "")
(while(/= nil (setq xs1WW (read-line fiz)))
(setq xs2WW (read-line fiz))
(setq xs3WW (read-line fiz))
```

```
(setq ris1WW (list ris1WW xs1WW))
(setq ris2WW (list ris2WW xs2WW))
(setq ris3WW (list ris3WW xs3WW))
)
```

```
(ZIZB2)
```

```
(setq ris1WW nil)(setq ris2WW nil)(setq ris3WW nil)
```

```
(setq xuks1WW nil)(setq xuks2WW nil)(setq xuks3WW nil)
(setq luks1WW nil)(setq luks2WW nil)(setq luks3WW nil)
```

```
(PORED)
```

```
(setq spaj1_ "lim.vi1")
(setq spaj2_ "lim.vi2")
(SPAJDA)
```

```
)
```

```
(defun SPAJDA())
```

```

(if(= nil (findfile (strcat "" spaj1_))) (progn
  (setq trw (open spaj1_ "w")) (close trw) ) (progn))
(if(/= nil (findfile (strcat "" spaj2_)))
  (progn
    (setq trw (open spaj1_ "a"))
    (setq trw2 (open spaj2_ "r"))
    (while(/= nil (setq nadjWWW (read-line trw2)))
      (write-line nadjWWW trw)
    )
    (close trw2)
    (close trw) (setq nadjWWW nil)
    (setq GHJJww nil)
  ) (PROGH))
(setq spaj1_ nil) (setq spaj2_ nil)
)
;-----new zizb

(defun pocpor()
  (while(/= guks1WWW "")
    (setq xuks1WWW (last guks1WWW))
    (setq guks1WWW (nth 0 guks1WWW))
    (setq luks1WWW (list luks1WWW xuks1WWW))
    (setq xuks2WWW (last guks2WWW))
    (setq guks2WWW (nth 0 guks2WWW))
    (setq luks2WWW (list luks2WWW xuks2WWW))
    (setq xuks3WWW (last guks3WWW))
    (setq guks3WWW (nth 0 guks3WWW))
    (setq luks3WWW (list luks3WWW xuks3WWW))
  )
)

(defun pored()

  (setq prosWWW "lim")

  (setq vik2WWW (strcat prosWWW ".vi2"))
  (setq vi2 (open vik2WWW "w"))

  (setq datsiWWW (strcat "sifr." prosWWW))

  (UPORED)

  (close vi2)

)

(defun upored()

  (setq RIWW " ") (setq ijWWW " ") (setq njWWW " ")
  (while(/= xis1WWW "")

```

```

(setq RIWW (last XIS1WW))(SETQ njWW (LAST XIS2WW))(SETQ ijWW (LAST XIS3WW))
(if(/= XIS1WW "")(progn
  (setq XIS3WW (nth 0 XIS3WW))
  (SETQ XIS2WW (HTH 0 XIS2WW))
  (SETQ XIS1WW (HTH 0 XIS1WW))
  )(progn))
(setq n1WW " ")(setq i1WW " ")(setq n2WW " ")(setq i2WW " ")
(setq njWW (atoi njWW))(setq ijWW (atoi ijWW))
(setq absi (open datsiWW "r")) (setq nadjWW "HEnadjen")
(while(and(/= nil (setq n1WW (read-line absi)))(/= nadjWW nil))
  (setq i1WW (read-line absi))
  (setq n2WW (read-line absi))
  (setq i2WW (read-line absi))

(setq n2WW (atoi n2WW))(setq n1WW (atoi n1WW))
(setq i1WW (atoi i1WW))(setq i2WW (atoi i2WW))

(if(and(> njWW n1WW)(< njWW n2WW))
  (progn
    (OTDALE)

  )(progn))
(if(and(= njWW n1WW)(>= ijWW i1WW))
  (progn
    (OTDALE)

  )(progn

  (if(and(= njWW n2WW)(<= ijWW i2WW))
    (progn

      (OTDALE)
    )(progn))
  ))

)
(close absi)
)

)

(DEFUH OTDALE()

  (setq n1carWW (itoa n1WW))(setq i1carWW (itoa i1WW))
  (setq s1WW (strcat n1carWW "-" i1carWW "." prosWW))

  (setq slb (open s1WW "r"))
  (dale) (close slb)
)

```

```

(DEFUH AKOIST()

(SETQ SVIBWW "--+0123456789.")(SETQ HKRWW (STRLEH RWW))(SETQ HQWW 1)(SETQ JSVIWW 0)
(WHILE(<= HQWW HKRWW)(SETQ QQZG (SUBSTR RWW HQWW 1))
(SETQ IQWW 1)
(WHILE(<= IQWW 13)(SETQ AZ1WW (SUBSTR SVIBWW IQWW 1))
(IF(= AZ1WW QQZG)(PROGH(SETQ JSVIWW (+ JSVIWW 1)))(PROGH))
(SETQ IQWW (+ IQWW 1)))
(SETQ HQWW (+ HQWW 1))
)
(IF(= JSVIWW HKRWW)
(PROGH(AKOIST1))(PROGH))
)
(DEFUH AKOIST1()
(setq jsvi1WW 0)          (setq ipWW 0)(setq itWW 0)(setq imWW 0)
(SETQ HKR1WW (STRLEH RIWW))(SETQ HQWW 1)
(WHILE(<= HQWW HKR1WW)(SETQ QQZG (SUBSTR RIWW HQWW 1))
      (if(= QQZG ".")(progn(setq itWW (+ itWW 1)))(progn))
      (if(= QQZG "+")(progn(setq ipWW (+ ipWW 1)))(progn))
      (if(= QQZG "-")(progn(setq imWW (+ imWW 1)))(progn))
(SETQ IQWW 1)
(WHILE(<= IQWW 13)(SETQ AZ1WW (SUBSTR SVIBWW IQWW 1))
(IF(= AZ1WW QQZG)(PROGH(SETQ JSVI1WW (+ JSVI1WW 1)))(PROGH))
(SETQ IQWW (+ IQWW 1)))
(SETQ HQWW (+ HQWW 1))
)
)
)

```

```

;---

```

```

(defun prstk()

(setq nan_ 0)
(setq una_ (length kcolb_)) (setq zg_ nil)
(setq zg_ 0)
(while(<= nan_ (- una_ 1))
...
...
...
)

```