Министерство образования Республики Беларусь

Учреждение образования
«Гомельский государственный технический
университет имени П. О. Сухого»

Кафедра «Технология машиностроения»

Н. А. Старовойтов, Е. Э. Дмитриченко

ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ ВЫСОКОГО УРОВНЯ GTL ПРИ РАЗРАБОТКЕ УПРАВЛЯЮЩИХ ПРОГРАММ ДЛЯ СВЕРЛИЛЬНО-ФРЕЗЕРНО-РАСТОЧНЫХ СТАНКОВ С ЧПУ

ПРАКТИКУМ
по дисциплине «Технология обработки
на станках с ЧПУ» для студентов специальности
1-36 01 01 «Технология машиностроения»
дневной и заочной форм обучения

Гомель 2018

УДК 621.9.06-529(075.8)
ББК 34.630.2я73
С77

Рекомендовано научно-методическим советом
машиностроительного факультета ГГТУ им. П. О. Сухого
(*протокол № 10 от 16.05.2017 г.*)

Рецензент: канд. техн. наук, доц. каф. «Сельскохозяйственные машины»
ГГТУ им. П. О. Сухого *А. В. Голопятин*

Старовойтов, Н. А.

С77 Программирование на языке высокого уровня GTL при разработке управляющих программ для сверлильно-фрезерно-расточных станков с ЧПУ : практикум по дисциплине «Технология обработки на станках с ЧПУ» для студентов специальности 1-36 01 01 «Технология машиностроения» днев. и заоч. форм обучения / Н. А. Старовойтов, Е. Э. Дмитриченко. – Гомель : ГГТУ им. П. О. Сухого, 2018. – 51 с. – Систем. требования: PC не ниже Intel Celeron 300 МГц ; 32 Mb RAM ; свободное место на HDD 16 Mb ; Windows 98 и выше ; Adobe Acrobat Reader. – Режим доступа: https://elib.gstu.by. – Загл. с титул. экрана.

Изложены различные варианты программирования на языке GTL составляющих элементов профилей обрабатываемых деталей: прямых, дуг, окружностей, способы обработки наружных и внутренних закрытых и открытых профилей с учетом коррекции диаметра режущего инструмента.

Для студентов специальности 1-36 01 01 «Технология машиностроения» дневной и заочной форм обучения..

УДК 621.9.06-529(075.8)
ББК 34.630.2я73

Введение

С момента появления первых станков с ЧПУ до внедрения новейших обрабатывающих центров появились различные языки программирования для создания управляющих программ (УП) при обработки. Сегодня программирование в G- и M-кодах является наиболее популярным. Язык G- и M-кодов основывается на положениях Международной организации по стандартизации (ISO) и Ассоциации электронной промышленности (EIA). Официально этот язык считается стандартом для американских и европейских производителей оборудования с ЧПУ, и иногда его называют «ИСО 7бит». Однако производители систем ЧПУ хоть и придерживаются этих стандартов для описания основных функций, но допускают вольности и отступления от правил, когда речь заходит о каких-либо специальных возможностях своих систем.

Разработано более 200 языков программирования для станков с ЧПУ, однако до сих пор не существует языка, который в полной мере удовлетворял бы всем требованиям

Языки отличаются:

-степенью специализации (универсальные и специальные для отдельных видов станков или типов деталей);

-степенью автоматизации технологических решений (не автоматизирующие технологию, частично и полностью автоматизирующие).

Различают: табличный и текстовой способы представления информации. В первом случае вся исходная информация задается в виде таблицы, во втором случае - обычным текстом.

В каждом языке различают:

-алфавит,

-синтаксис,

-семантику.

Алфавитом называют множество символов, которые используют для обозначения сообщений. Обычно он содержит цифры от 0 до 9, буквы латинского или русского алфавита и другие знаки (плюс, минус, скобки, точка, запятая и т. д.).

Синтаксис излагает правила, по которым из символов алфавита можно формировать языковые конструкции (слова), какие сочетания символов допустимы, какие нет.

Семантика определяет смысловое содержание слов. Каждую конструкцию языка нужно толковать однозначно и определенно.

Основу любого языка программирования составляют способы определения геометрических элементов:

-точки,

-прямой,

-окружности (дуги).

1.Геометрическое программирование высокого уровня на языке (GTL)

В данном пособии речь пойдет об одном из языков программирования **GTL**, основанном на векторной геометрии, весьма простом, легким в освоении и применении. Данный язык применяется в системах ЧПУ для ряда моделей серии NC ООО «БалтСистем» РФ, продажа которых на рынках СНГ в настоящее время доминирует.

К G-функциям определяющим запрограммированный геометрический профиль с использованием языка GTL, принадлежат две функции:

G21 - устанавливает начало геометрического профиля на базе языка GTL;

G20 - устанавливает конец геометрического профиля на базе языка GTL.

В системе УЧПУ серии NC представляется возможным в программе описать геометрический профиль в плоскости, используя не только стандартный язык программирования (G1-G2-G3), но и язык программирования высокого уровня GTL. Этот язык позволяет программировать профиль, состоящий из прямых и окружностей (дуг), используя только информацию, полученную с чертежа. Система сама вычисляет точки пересечения и точки касания геометрических элементов.

Язык программирования GTL и стандартный язык могут быть использованы одновременно в одной и той же программе, но не для одного и того же профиля. Геометрия GTL функционирует только при абсолютном программировании **(G90).**

1.1 Векторная геометрия

Определение профиля с использованием GTL основано на использовании четырёх типов геометрических элементов и обозначается строчными латинскими символами:

o-точки начала отсчёта;

p-точки;

l-прямые;

c-окружности.

Так как профиль определяется как геометрическими элементами, так и направлением движения по нём, то для определения геометрических элементов в языке GTL используется особый тип геометрии - ***векторная геометрия***. Для векторной геометрии определение элемента кроме параметров, необходимых для установления позиции в плоскости, требует также ***назначения направления движения.***

Например, прямая линия входит в обрабатываемый профиль и движение инструмента происходит от точки **A** к точке **B,** как показано на рис. 1.1, то ее направление будет по стрелке 1, а при движении от **B** к **A, - 1'**.

Рис.1.1.Определение прямых в векторной геометрии.

В векторной геометрии **1** и **1'** являются двумя различными линиями, имеющими противоположные направления. Программирование при помощи языка GTL, основанное на векторной геометрии, требует для каждой прямой линии назначения направления движения. Условимся, что направление движения по прямой определяется углом, который она образует с положительной осью X. Положительный угол получается при вращении положительной оси X против часовой стрелки до наложения его на одну из прямых линий, которую надо определить.

Угол будет иметь положительный знак плюс (не указывается), если ось X будет вращаться против часовой стрелки, и отрицательный минус (-) в обратном направлении, как показано на рис. 1.2.

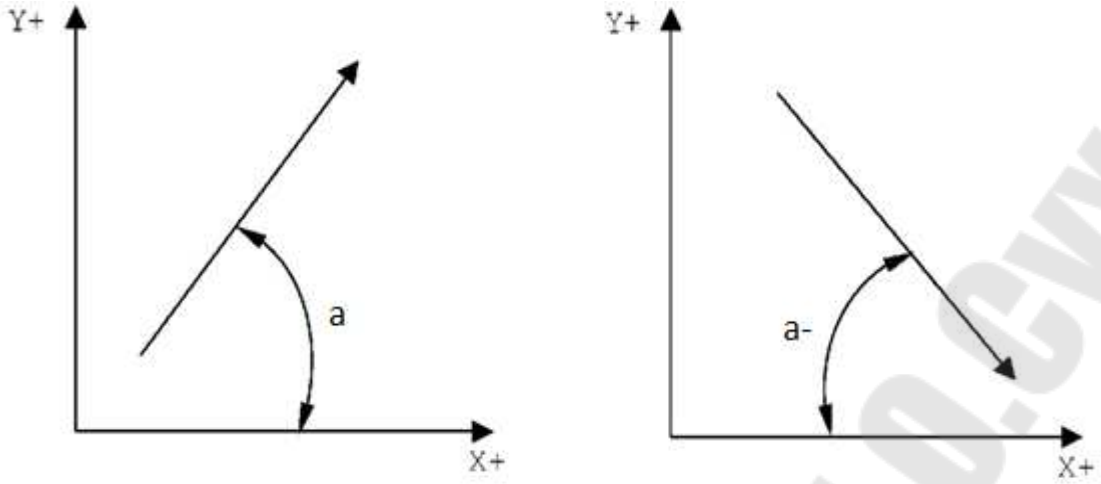

Рис.1.2. Определение угла при вращении прямой линии.

Направление движения инструмента должно быть придано также и окружностям. Условно принимается за положительное направление движение по окружности против часовой стрелки + плюс, (не указывается) и за отрицательное (-) минус по часовой стрелке, как показано на рис. 1.3.

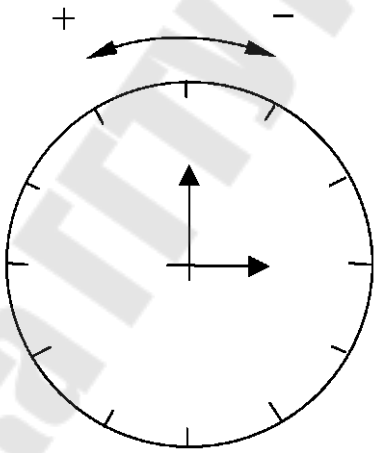

Рис. 1.3. Определение направление движения по окружности.

При определении на плоскости окружности (дуги) необходимо знать величину радиуса. Принято считать, что положительное значение радиуса (+) плюс (не указывается) придается окружностям с направлением движения против часовой стрелки и отрицательное (-) минус в обратном направлении, как показано на рис. 1.4.

Направление движения инструмента, обычно соответствует с направлением движения по профилю. Однако, можно изменить направление элемента во время определения профиля, если это направление противоположно остальным элементам профиля путем изменения

знака перед буквой определяющей профиль, например, -l на l; c на –c; r на –r; и т.д., если это предусмотрено форматом программирования.

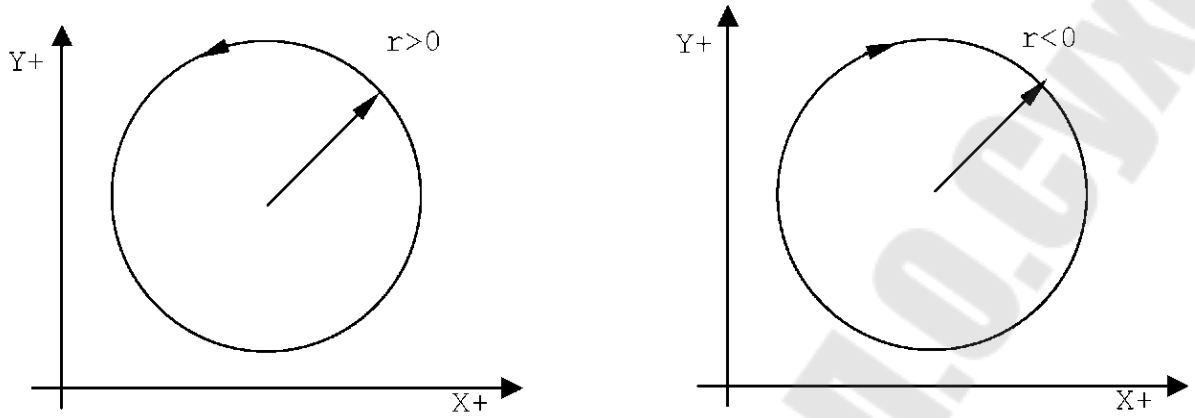

Рис. 1.4. Определение значения радиуса в зависимости от направления движения по окружности.

1.2. Хранение в памяти геометрических элементов

Хранение в памяти геометрических элементов предусматривает использование строчных символов **a; l; c; d; m; o; r; p; s; b** для определения соответственно:

a - углов;
l - прямых линий;
c - окружностей;
d - расстояний;
m - модулей;
o - точек начала отсчета;
r - радиуса;
p - номер пересечений,
s - дискриминатор;
b - скоса.

Необходимость использования для этой информации строчных символов вызвана тем, что эти же ***заглавные символы*** *используются в языке ЧПУ для другой и*нформации. ***Запоминание геометрических элементов в памяти осуществляется до определения профиля.*** Элементами, рассматриваемыми в языке GTL, являются прямые, окружности, точки, точки начала отсчета. Это - геометрические переменные, идентифицированные **НАЗВАНИЕМ** и **ИНДЕКСОМ.** Геометрическая переменная определяется в кадре назначения.

Формат:

НАЗВАНИЕ ИНДЕКС=<выражение>,

где:

НАЗВАНИЕ - одно из четырёх символических названий, предусмотренных для геометрических элементов;

o - для определения точки начала отсчета;

p - для определения точки при пересечении элементов;

l - для определения прямой;

c - для определения окружности;

ИНДЕКС - определяет номер переменной геометрического элемента. Этот номер заключён между 0 и 255, максимальный предел определяется при характеризации ЧПУ.

выражение - содержит всю информацию, необходимую для описания геометрического элемента; элементы могут быть определены:

- явным образом, программируя в кадре всю информацию, необходимую для определения геометрического элемента;
- неявным образом, вызывая другие геометрические элементы, определённые ранее.

Пример хранения элементов в памяти.

Число геометрических элементов, хранимых в памяти, определяется на стадии характеризации системы. Формат геометрических определений предусматривает использование символа «,» (запятая) для отделения геометрического элемента (прямая - точка - окружность) от последующего геометрического элемента или информации (такой, как радиус «**r**» или угол «**a**»).

Примеры

p1=X30Y30 – разделитель (запятая) между координатами не требуется

c1=I10J20r30 – разделитель (запятая) между координатами не требуется

l1=X20Y20,X100Y-10 - разделитель (запятая) требуется между координатами первой точки (X20 Y20) и координатами второй точки (X100Y-10)

l2=I30J20r10,X80Y80 - разделитель (запятая) требуется между координатами окружности (I30J20r10) и координатами точки (X80Y80).

l3=X100Y100,a45 - разделитель (запятая) требуется между координатами точки (X100Y100) и углом (a45).

c3=l1,l2,r18 - разделитель (запятая) требуется между прямыми l1, l2 и радиусом (r18).

Дискриминатор **s2** служит для выбора второго пересечения (**s2**). Иллюстрация приведена на рисунке 1,5.

P1=c1,l3
P2=c1,l3,s2

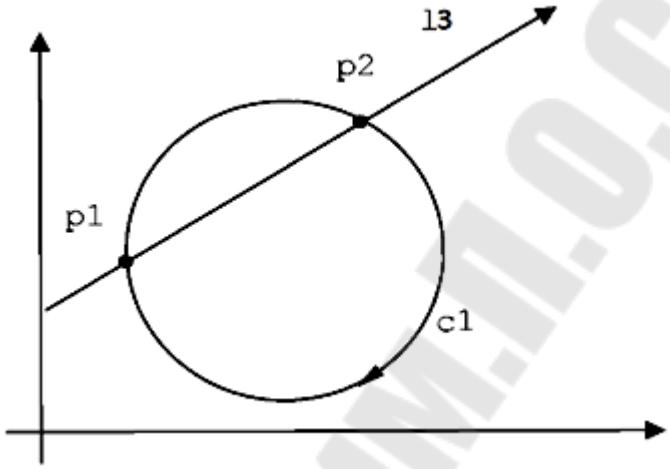

Рис.1.5.Определения дискриминатора **s**.

1.3 Определение точек начала отсчёта.

Функция определения точек начала отсчёта дает возможность определить точки начала отсчёта в ***прямом формате (явным образом).***

Обычно информация, находящаяся в программе, относится к систем осей, совпадающих с осями станка. Однако при проектировании деталь может быть выполнена на чертеже с использованием различных декартовых систем: абсолютной системы и других систем (начальных точек) отсчёта, которые могут быть приведены к абсолютной системе вращением и смещением осей. Геометрия GTL может быть определена при любой системе начала отсчёта.

Формат:

on=X..Y..a..

где:

on - определяет название точки начала отсчёта;

X..Y.. - координаты новой начальной точки;

a.. - угол вращения (положительный против часовой стрелки).

Пример приведён на рисунке 1.6.

Примечание:

Если после символов в написании формата следует двоеточие, то это означает, там могут быть переменные, например, числовые значения.

o4=X20Y30a-45

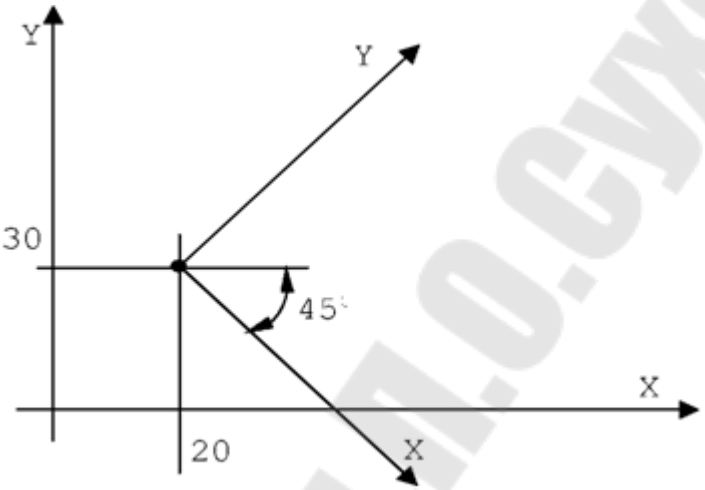

Рис.1.6.Определение точки начала отсчета.

1.4.Определение точек

Функция определения точек позволяет определить точки в прямой (явной) форме или в косвенной (неявной) форме. Определение может быть дано как в декартовых координатах, так и в полярных.

Система полярного начала отсчета состоит из начальной точки, называемой полюсом, из которой начинается ось X, называемая полярной осью. Система полярного начала отсчёта приведена на рис.1.7

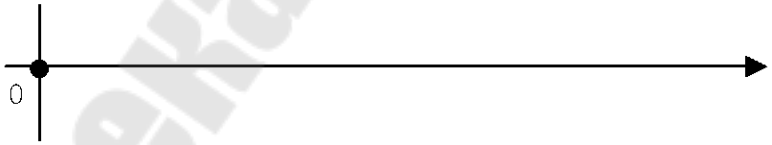

Рис.1.7 - Система полярного начала отсчёта.

В общем случае любая точка на плоскости может быть определена при помощи длины отрезка **p (** названной далее модулем ***m),*** который соединяет ее с полярной точкой P, и при помощи угла ϑ (тэта), который образуется отрезком прямой и полярной осью, как показано на рисунке 1.8.

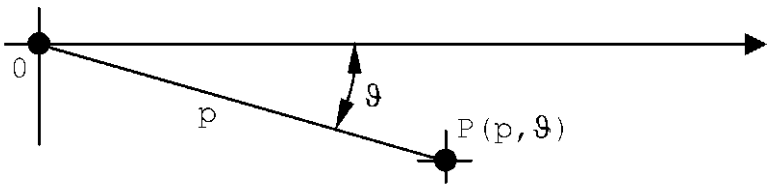

Рис.1.8. – Общий случай определения точки в полярных координатах.

Прямой (явный) способ определение точки в полярны координатах показан на рис.1.9:

Формат:

pn=[on]m..a..

Примечание:

1. Если в формате написания присутствуют квадратные скобки, то это значит, что значения в квадратных скобках может не быть или может быть альтернативным.

2.Прямой (явный) способ определения означает, что элемент контура определяется первоначально с использованием всех значений необходимых для этого координат.

3.Написание информации во всех форматах производится без пробелов.

p2=m55a60

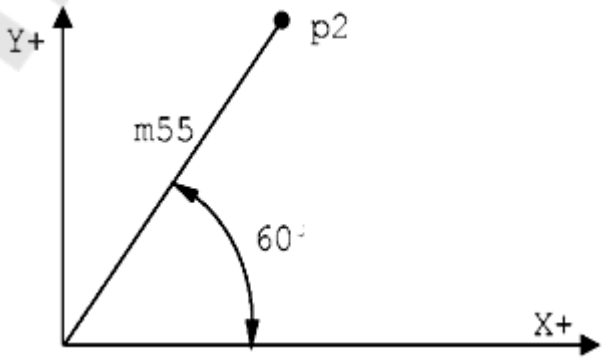

Рис.1.9. Прямой способ определения точки в полярных координатах.

1.Определение точки прямым способом в декартовых координатах показано на рис.1.10 и 1.11:

Формат:

pn=[on]X..Y..

Пример прямого способа определения точки в системе декартовых координатах (рис.1.10 и 1.11.

p1=X30Y160

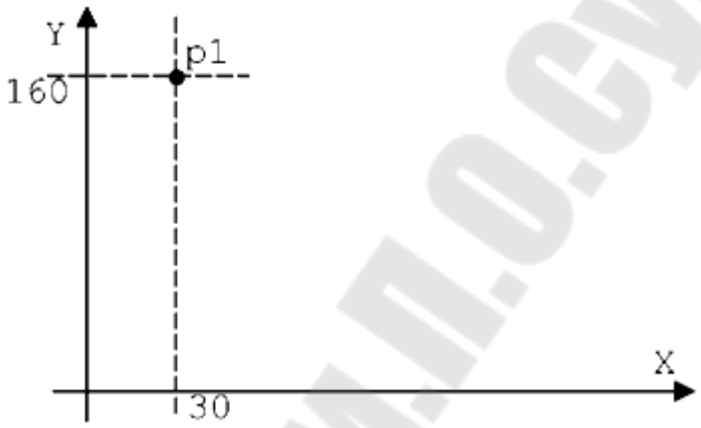

Рис.1.10. Прямой способ определения точки без смещения начальной точки системы координат/

o1=X30Y30A-20
p5=o1X20Y10

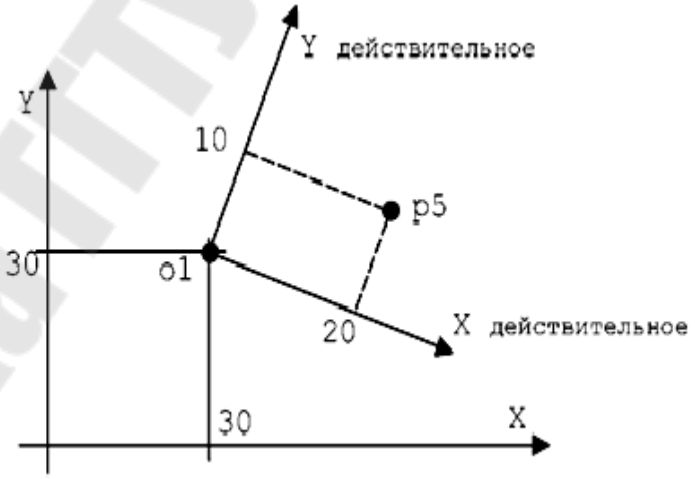

Рис.1.11. Прямой способ определения точки в смещенной системе координат.

Пример косвенного (неявного) способа определения точки в декартовых координатах с помощью двух прямых, определённых ранее, показано на рис.1.12.

Формат:

pn=lm,lp

Примечание:

Косвенным (неявным) способом считается такой способ, когда для определения профиля ***используются элементы профиля ранее определенные.***

p1=l1,l2

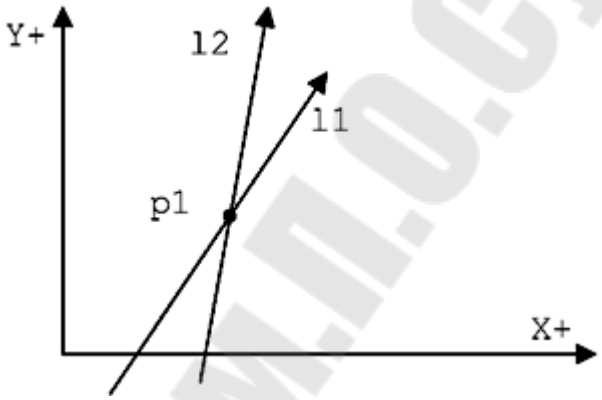

Рис.1.12. Пример косвенного способа определения точки.

Пример косвенного (неявного) способа определения точки при пересечении прямой и окружности, определённых ранее, приведен на рис. 1.13.:

Формат:

pn=[-]lm,cp[,s2]

pn=cm,[-]lp[,s2]

Примечание:

В форматах перед прямой **l**(эль) стоит в квадратных скобках знак минус:[-].Это значит, что с его помощью возможно поменять направление прямой на противоположное. В связи с этим точку **p1** можно определить двумя способами.

В случае пересечения прямая-окружность или наоборот, существуют два возможных решения: окружность **c3** и прямая **l4** пересекаются в точках **p1** и **p2**. Проходя прямую **l4**, следуя её направлению, сначала встречаем точку **p1** (1-е пересечение), а затем - точку **p2** (2-е пересечение). Для выбора второго пересечения (**p2)** следует использовать индикатор **s2.** Если он опущен, то выбирается первое пересечение (**p1).**

p1=l4,c2
p2=l4,c3,s2
p1=-l4,c3,s2

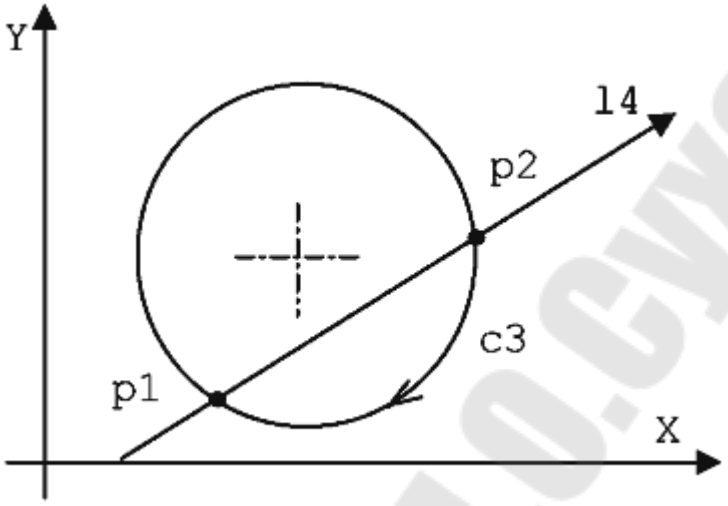

Рис.1.13. Пример косвенного способа определения точки при пересечении прямой и окружности.

Пример косвенного способа определения точки при пересечении двух окружностей показан на рис.1.14.:

Формат:

pn=cm,cp[,s2]

p1=c1,c2
p2=c1,c2,s2
p1=c2,c1,s2

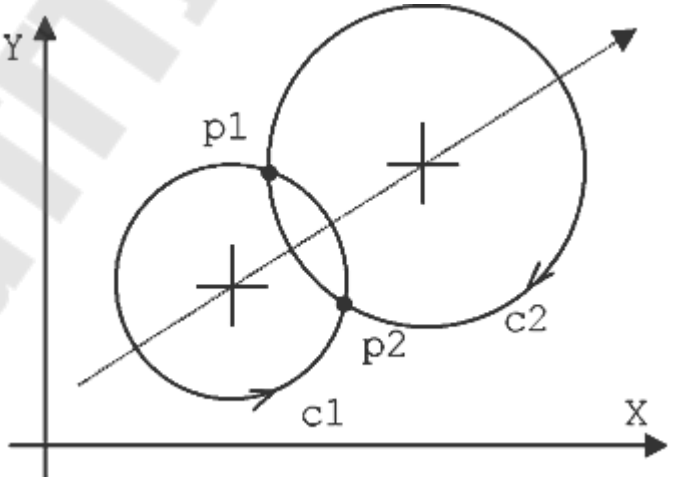

Рис.1.14. Пример косвенного способа определения точки при пересечении двух окружностей.

В случае пересечения окружность-окружность существуют два возможных решения:

- окружности **c1** и **c2** пересекаются в точках **p1** и **p2** (рис. 1.14). В этом случае рассматривается условно сориентированная прямая в направлении движения по профилю, соединяющая центр 1-ой окружно-

сти с центром 2-ой окружности. Она делит плоскость на две полуплоскости. Для выбора точки в правой полуплоскости (**p2)** следует использовать дискриминатор **s2.** Если он опущен, то автоматически выбирается точка в левой полуплоскости (**p1).**

Общие обозначения, встречающиеся в форматах:

pn - определяет название точки индекса **n** (**n** – порядковое число, заключённое между 1 и максимальным числом определенным при конфигурации системы ЧПУ);

X..Y.. - координаты точки;

[on] - начальная точка отсчёта индекса **n**, определённая ранее, к которой относятся координаты **X** и**Y**;

m - модуль полярного вектора;

a.. - угол полярного вектора;

cm,cp - ранее определённые элементы окружности индекса **m** и **p**; (**m** и **p** – порядковые числа, заключённые между 1 и максимальным числом определенным при конфигурации системы ЧПУ);

[-]lm - ранее определённые элементы прямой индекса **m**, можно изменить направление, вставляя знак «-»;

[-]lp - ранее определённые элементы прямой индекса **p**, можно изменить направление, вставляя знак «-»;

[,s2] – индикатор (дискриминатор) второго пересечения.

1.5.Определение прямой линии

Функция определения прямой линии позволяет определить прямую линию в прямой (явной) или косвенной (неявной) форме.

Направление прямой линии всегда от первого ко второму среди определяемых элементов. В случае если прямая касается с окружностью, возможны два решения, т.к. прямая может быть касательной с одной или с другой стороной окружности. Для выбора требуемого решения следует убедиться в том, что в точке касания окружность и прямая имеют одно и то же направление.

Несовместимость направлений движения геометрических элементов приведена на рисунке 1.15. Совместимость направления движения приведена на рисунке 1.16.

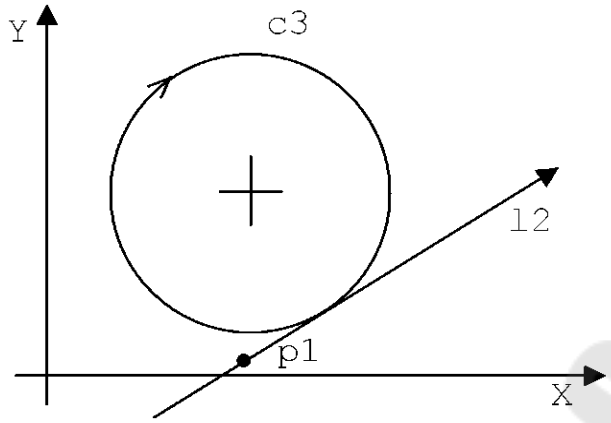

Рис.1.15. Несовместимость направлений движения геометрических элементов.

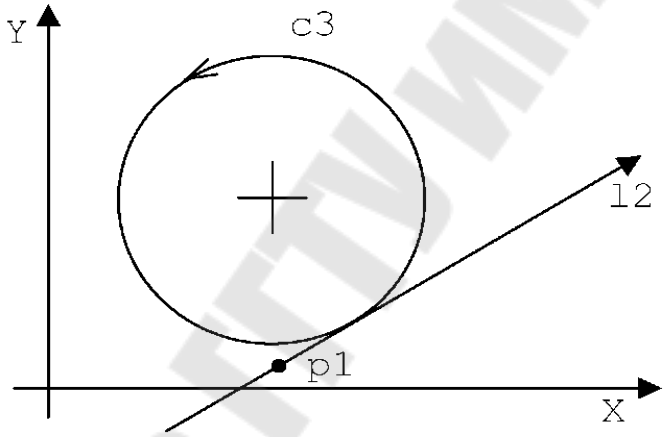

Рис.1.16 Совместимость направлений движения геометрических элементов.

1.5.1 Прямая форма (явная) программирования прямых линий:

Формат:

- прямая, проходящая через две точки (рис.1.17):

ln=[om]X..Y..,[op]X..Y..

l1=X40Y20,X60Y70

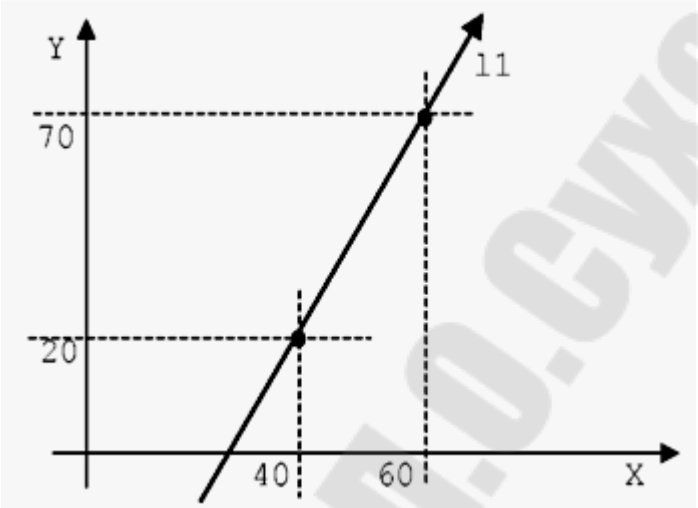

Рис.1.17. Определение прямой, проходящей через две точки.

Формат:

- прямая, проходящая через одну точку и образующая угол с осью абсциссы (рис. 1.18):

ln=[om]X..Y..,a..

l2=X20Y20,a-20

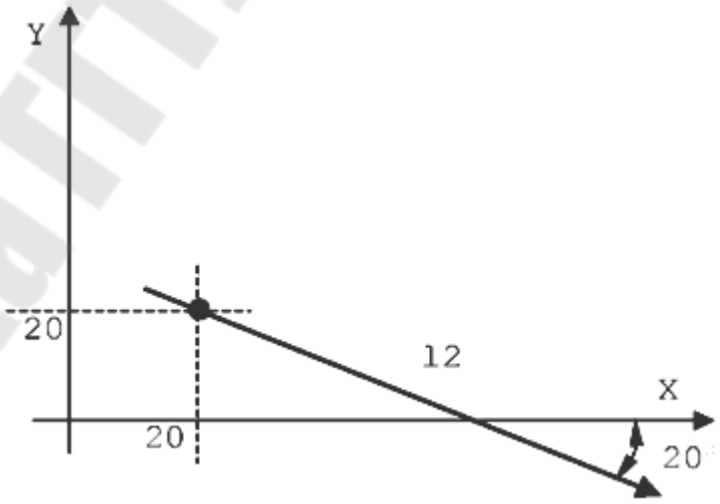

Рис. 1.18. Определение прямой, проходящей через одну точку и образующей угол с осью абсциссы.

Формат:

- прямая, проходящая через одну точку и образующая угол с осью абсциссы в смещенной системе координат (рис.1.19):

ln=[om]X..Y..,a..

o1=X30Y30a-40
l5=o1X25Y30,a60

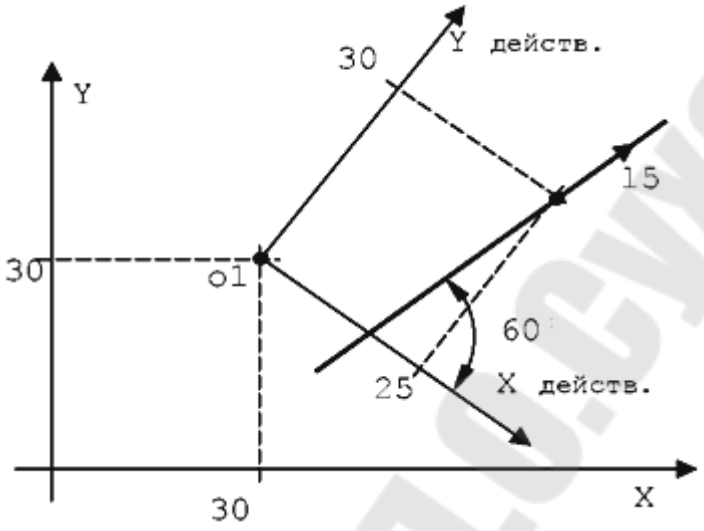

Рис. 1.19. Определение прямой, проходящей через одну точку и образующей угол с осью абсциссы в смещенной системе координат.

l1=I60J80,a45

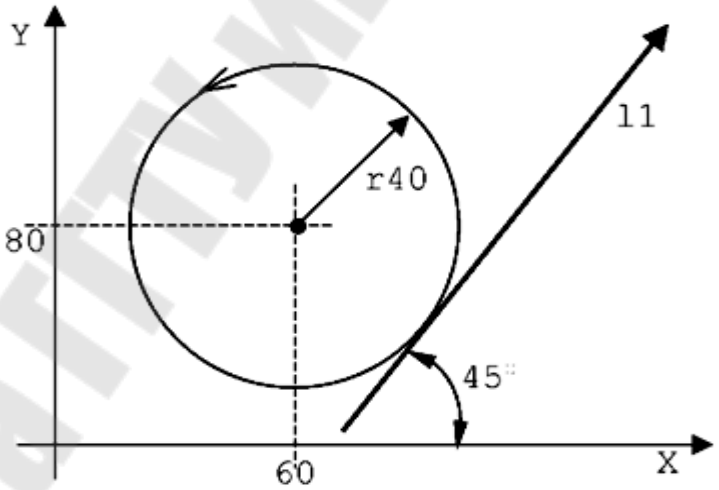

Рис.1.20. Определение прямой, касательной к одной окружности и образующей угол с осью абсциссы.

Формат:

- прямая, касательная к одной окружности и образующая угол с осью абсциссы в смещенной системе координат (рис. 1.21).

ln=[om]I..J..r..,a..

o3=X25Y10a20
l4=o3I25J15r10,a115

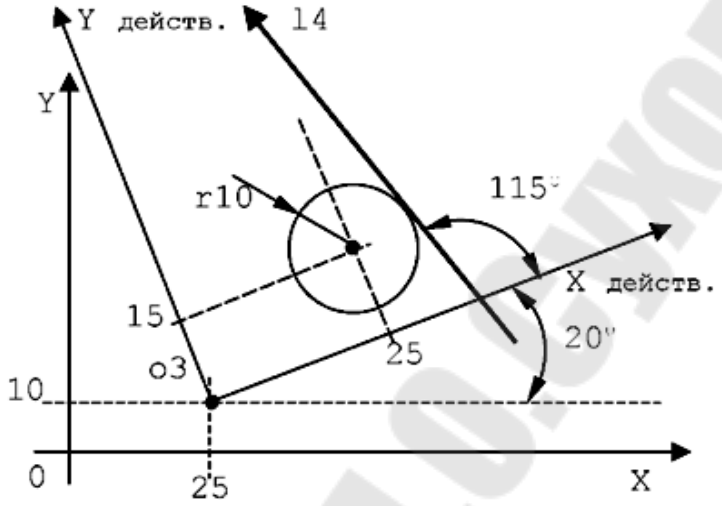

Рис.1.21.Определение прямой, касательной к одной окружности и образующей угол с осью абсциссы в смещенной системе координат.

Формат:
- прямая, касательная к двум окружностям (рис.1.22):
ln=omI..J..r..,opI..J..r..

l3=I25J35r-17,I70J20r13

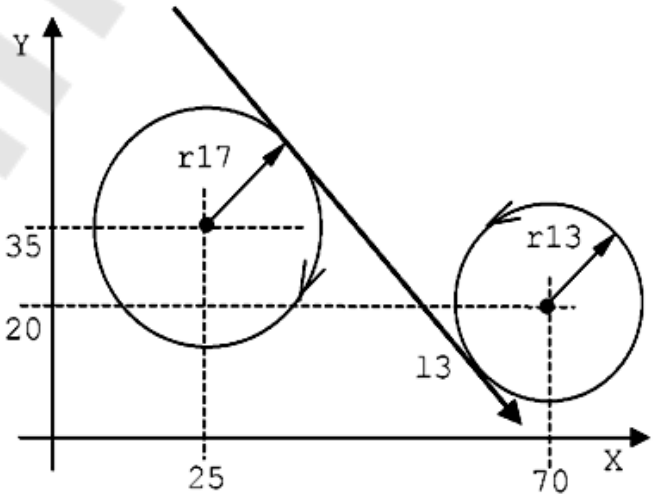

Рис.1.22. Определение прямой, касательной к двум окружностям.

Формат:
- прямая, касательная к двум окружностям (рис.1.23):
ln=omI..J..r..,opI..J..r..

l4=I25J35r17,I70J20r13

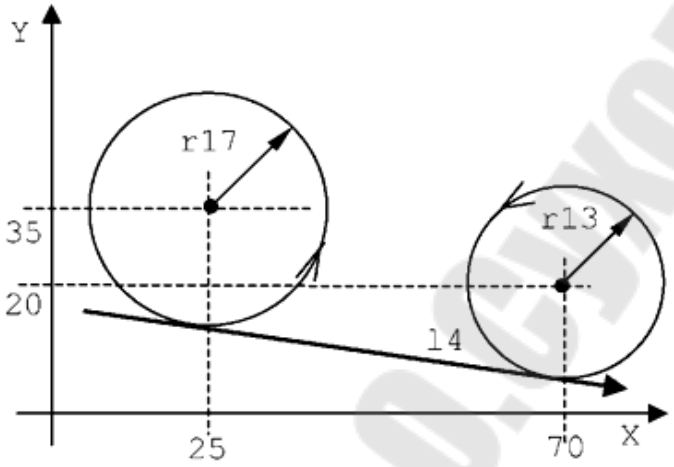

Рис. 1.23. Определение прямой, касательной к двум окружностям.

Формат:

- прямая, касательная к окружности и проходящая через точку (рис. 1.24):

ln=[om]I..J..r..,[op]X..Y..

ln=[om]X..Y..,[op]I..J..r..

l1=X10Y15,I45J30r-15

l2=X10Y15,I45J30r15

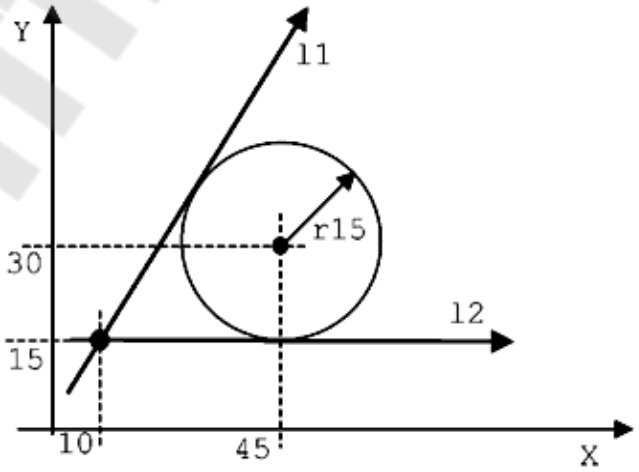

Рис.1.24.Определение прямой, касательной к окружности и проходящей через точку

1.5.2.Косвенная форма(неявная) программирования прямых линий:

Формат:

- прямая, проходящая через две точки (рис.1.25):

ln=pm,pq

l9=p7,p8

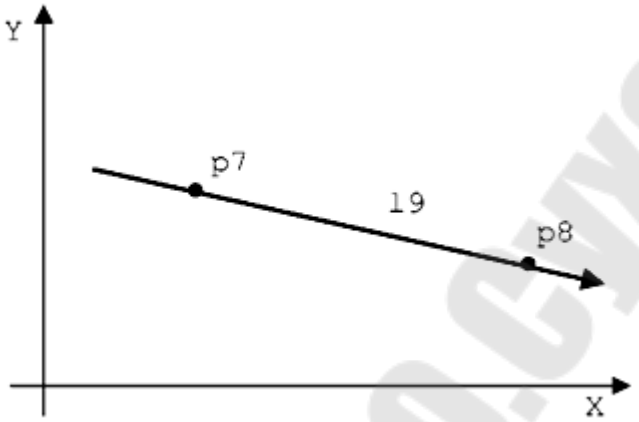

Рис. 1.25. Определение прямой, проходящей через две точки.

Формат:

- прямая, проходящая через точку и образующая угол с осью абсциссы (рис. 1.26):

ln=pm,a..

l3=p1,a45

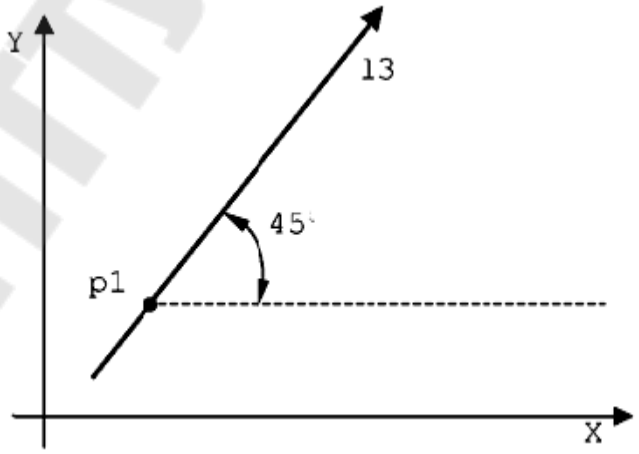

Рис.1.26. Определение прямой, проходящей через точку и образующей угол с осью абсциссы.

Формат:

- прямая, касательная к двум окружностям (рис.1.27,1.28):

ln=[-]cm,[-]cp

l3=c1,c2

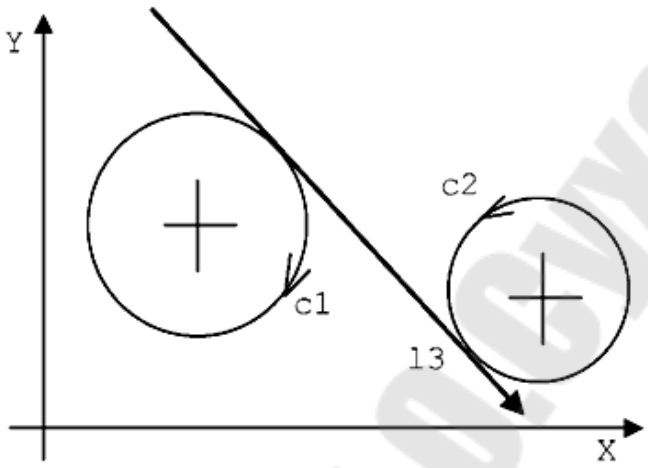

Рис.1.27.Определение прямой, касательной к двум окружностям.

l4=-c1,c2

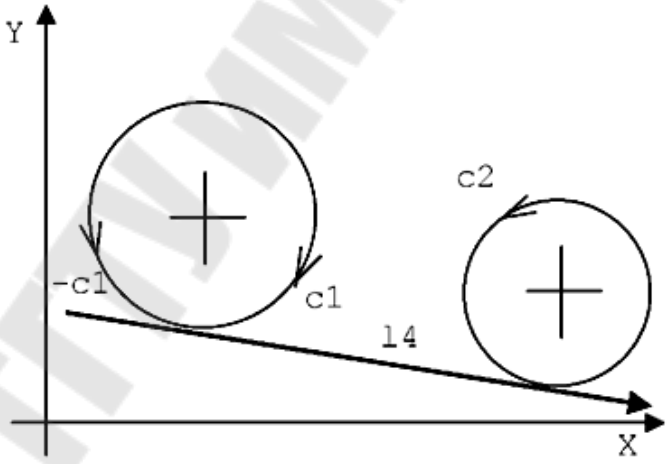

Рис.1.28.Определение прямой, касательной к двум окружностям.

Формат:

- прямая, касательная к одной окружности и образующая угол с осью абсциссы (рис.1.29):

ln=[-]cm,a..

l1=c1,a50
l2=-c1,a50

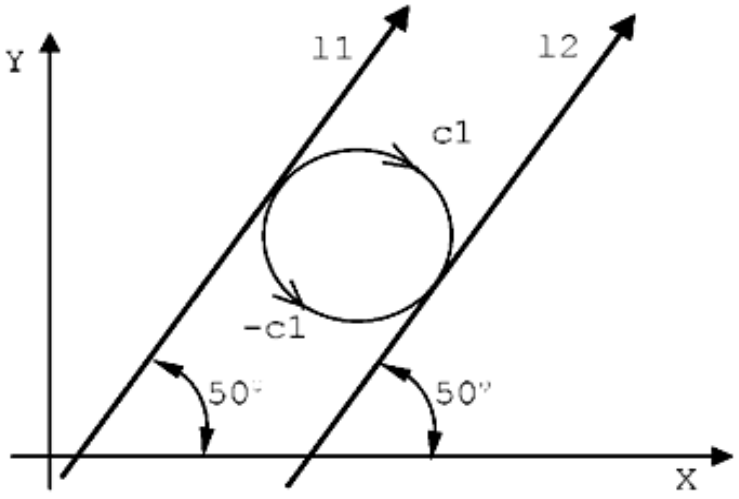

Рис 1.29.Определение прямой, касательная к одной окружности и образующая угол с осью абсциссы.

Формат:

- прямая, касательная к окружности и проходящая через точку (рис.1.30):

ln=[-]cp,pm

ln=pm,[-]cp

l1=p1,c1

l2=p1,-c1

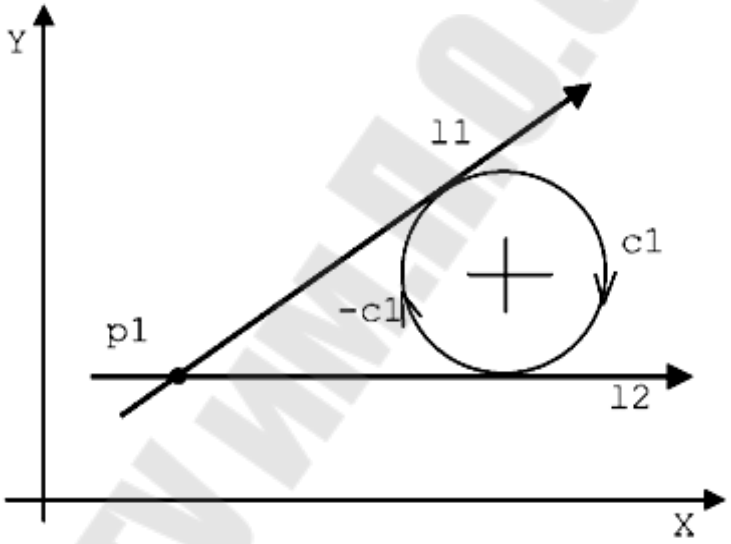

Рис.1.30.Определение прямой, касательная к окружности и проходящая через точку.

Формат:

- прямая, параллельная прямой на расстоянии d (рис. 1.31, 32):

ln=[-]lm,d..

l2=l1,d20

l3=l1,d-15

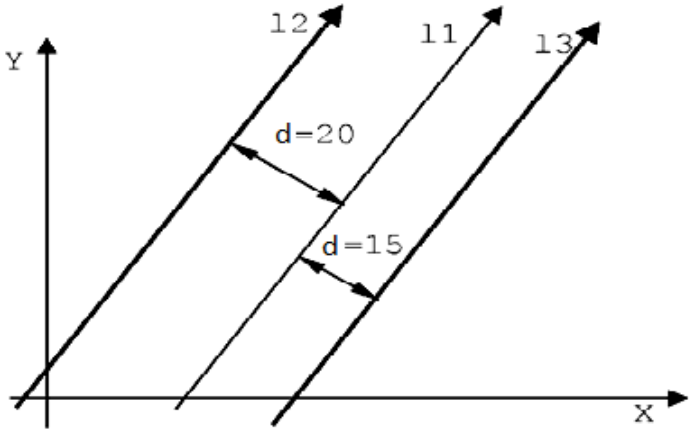

Рис.1.31.Определение прямой, параллельная прямой на расстоянии d.

l2=-l1,d-50

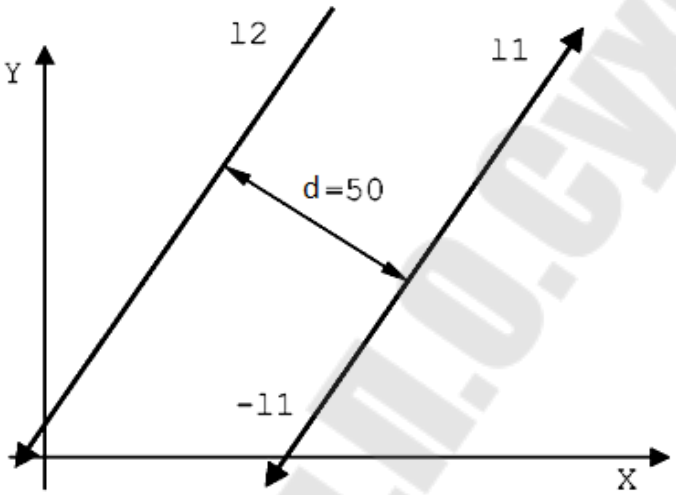

Рис.1.32. Определение прямой, параллельная прямой на расстоянии d.

Общие обозначения, встречающиеся в форматах:

ln - определяет название прямой с индексом **n** (**n** – порядковое число, заключённое между 1 и максимальным числом определенным при конфигурации системы ЧПУ);

X..Y.. - координаты точки;

a.. - угол, образованный осью абсциссы и прямой (положительный отсчитывается против часовой стрелки);

r.. - радиус окружности (положительный при движении против часовой стрелки и отрицательный со знаком минус - по часовой стрелке);

pm, pq - предопределённые элементы точки с индексами **m** и **q;** (**m** и **q** – порядковые числа, заключённые между 1 и максимальным числом определенным при конфигурации системы ЧПУ);

[-]cm[-]cp - предопределённые элементы окружности с индексами **m** и **p.**(**m** и **p** – порядковые числа, заключённые между 1 и максимальным числом определенным при конфигурации системы ЧПУ).

Направление движения по окружности может быть изменено при помощи отрицательного знака для гарантирования совместимости направлений прямой и окружности в точке касания, на это указывает в формате знак **[-].**

[-]lm - предопределенный элемент прямой с индексом **m**;

d.. - расстояние между двумя прямыми: положительное, если прямая находится слева, отрицательное – если справа, ***глядя в направлении, предопределенном прямой.***

1.6.Определение окружностей

Язык GTL позволяет определить окружности в прямой форме (явной) или в косвенной (неявной) форме. Определяя окружности в косвенной форме, программист должен учитывать совместимость направлений элементов (знак «-»(минус) может изменить направление предопределенных элементов). Если не учитывается направление элементов, то задавая окружность известного радиуса и прямую линию, можно получить от 1 до 8 вариантов окружности, касательной к двум элементам. Возможные варианты окружности, касательной к прямой и окружности приведены на рисунке 1.32.

Если учитывать совместимость направлений движения предопределённых элементов и окружности, которую следует определить, количество возможных решений можно уменьшить до двух.

Чтобы различить две возможные окружности, имеющие одно и то же направление и один и тот же радиус, необходимо учитывать последовательность элементов:

- прямая - окружность (окружность - прямая);
- значение центрального угла двух возможных тангенциальных окружностей.

GTL всегда создает окружность с направлением от первого ко второму элементу и дугой, имеющей меньший центральный угол.

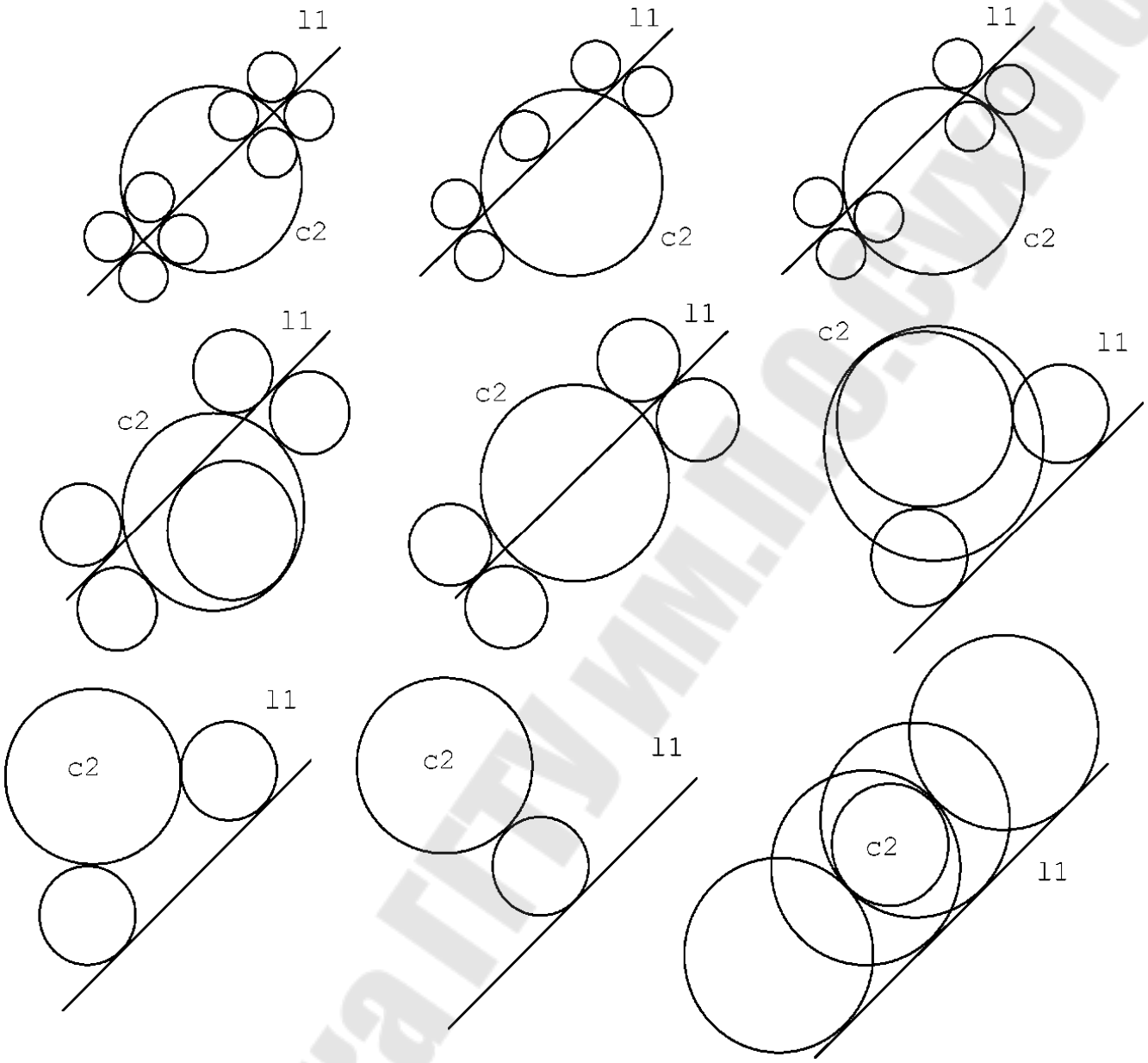

Рис.1.33.Различные варианты окружности касательные к двум элементам.

Окружности, касательные к меньшей дуге, показаны на рис.1 .33. Окружность **c3** получается при определении прямой **l1** в первой позиции и окружности **c2** - во второй, т. к. элемент **c3** позволяет движение от прямой **l1** к окружности **c2** с дугой, имеющей меньший центральный угол α.

Окружность **c4** получается при определении окружности **c2** в первой позиции и прямой **l1** - во второй, т.к. элемент **c4** позволяет движение от окружности **c2** к прямой **l1** с дугой, имеющей меньший центральный угол γ.

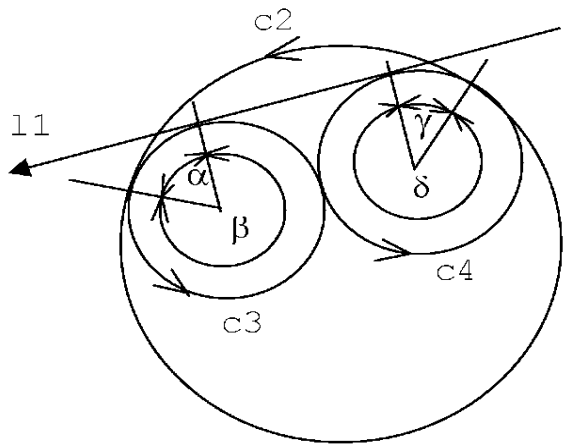

Рис. 1.34. Пример окружностей, касательных к меньшей дуге.

То же самое можно сказать для определения окружности, касательной к двум предопределенным окружностям. В этом случае можно получить от 1 до 8 решений в соответствии с рис.1.33 Окружности, касательные к двум предопределенным окружностям, показаны на рис.1.35.

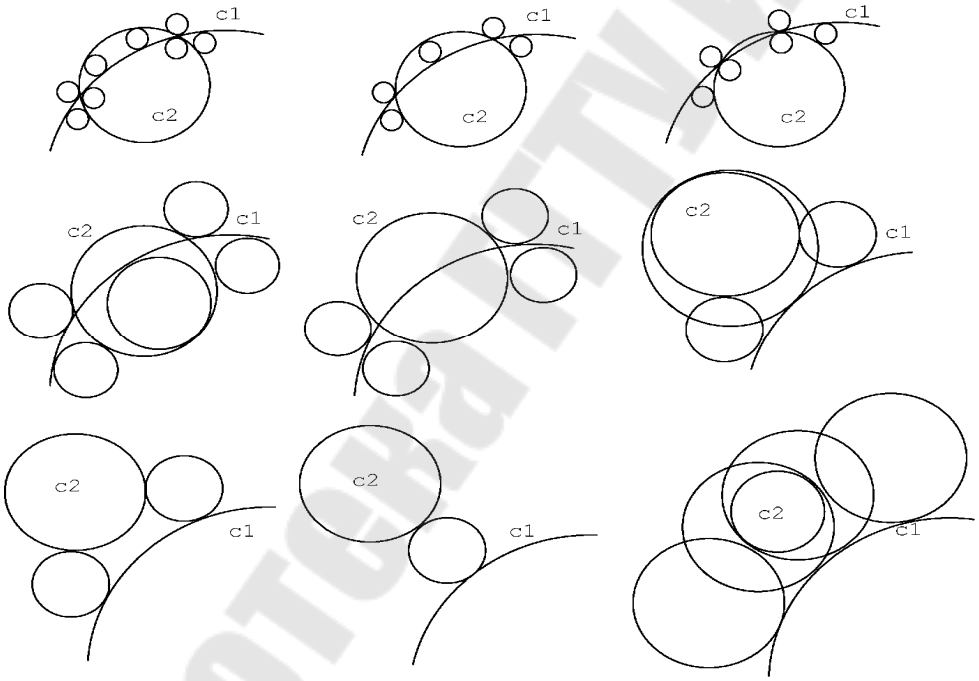

Рис.1.35.Различные варианты окружностей, касательных к двум предопределенным окружностям.

Совместимость направлений движения предопределенных элементов и окружности, которую надо определить, сводит количество возможных решений до двух. Для различения двух окружностей, имеющих одинаковое направление и одинаковый радиус, рассматри-

ваются две дуги с углом α и γ, в которых новый элемент разделен точками, касательными к элементам начала.

GTL создает окружность, двигаясь от первой окружности ко второй, с дугой α и γ, имеющими меньший центральный угол (рис.1.36). ***Для получения окружности c3 при определении следует установить в правой позиции элемент c1, а затем c2. Для получения окружности c4 элемент c2 должен предшествовать элементу c1 в определении.***

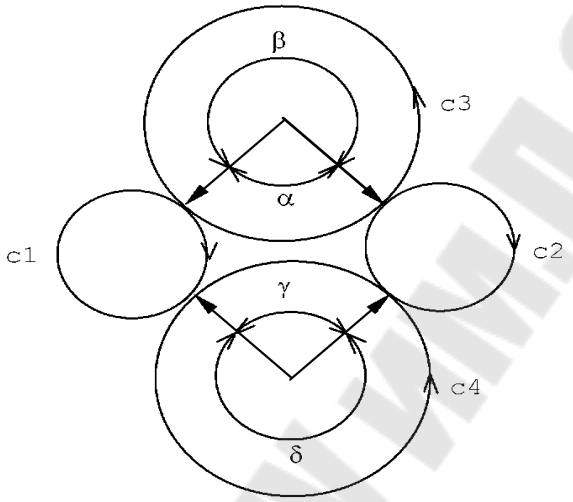

Рис. 1.36. Определение окружностей, касательных по меньшей дуге к двум предопределенным окружностям.

1.6.1.Формат прямого программирования окружностей

Формат:

- окружность с декартовыми координатами центра и радиусом (рис.1.37;1.38):

cn=[om]I..J..r..

c2=I50J100r-40

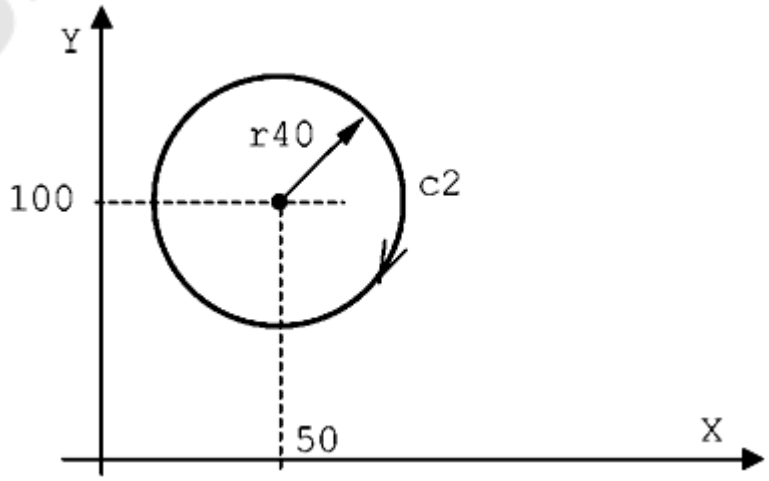

Рис.1 37.Определение окружности с декартовыми координатами центра и радиусом.

c1=o1I20J20r-15

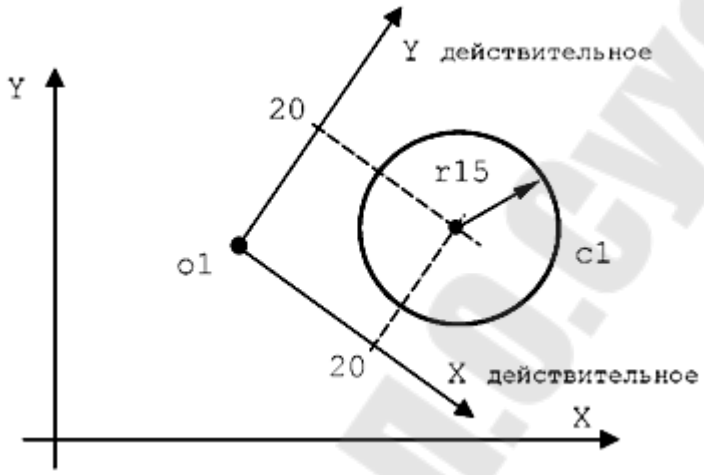

Рис.1.38. Определение окружности с декартовыми координатами центра и радиусом в смещённой системе координат.

Формат:

- окружность с полярными координатами центра и радиусом (рис.1.39):

cn=[om]m..a..r..

c2=m70a30r15

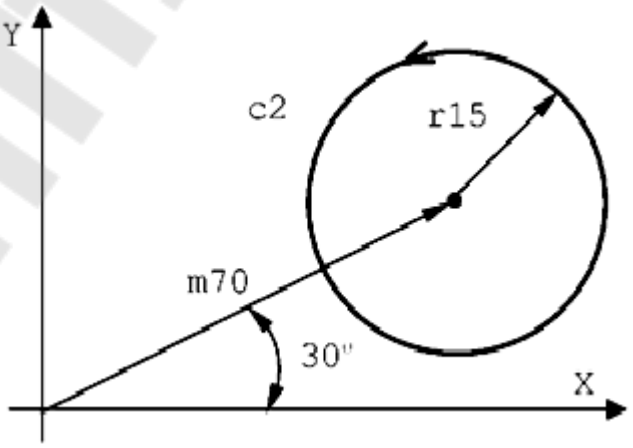

Рис.1.39. Определение окружности с полярными координатами центра и радиусом.

Формат:

cn=I..J..r..,d.. - расстояние между двумя окружностям (рис.1.40)

c6=I50J40r20,d-10

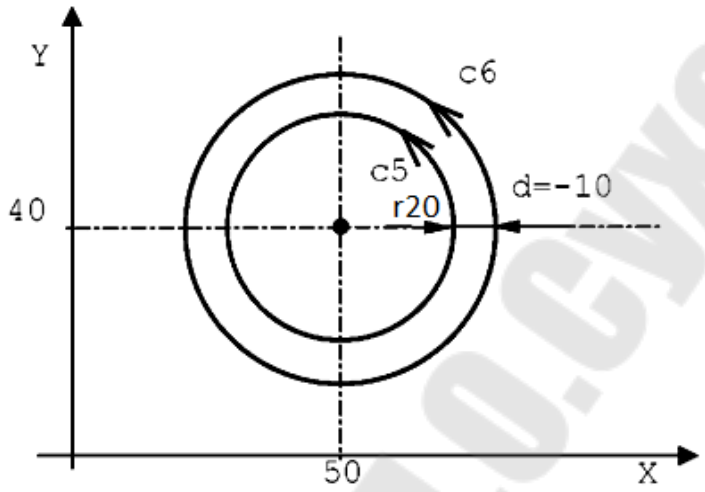

Рис.1.40. Определение окружности заданием расстояния между двумя окружностями.

При использовании знака «-» минус, (**d=-10**), радиус новой окружности **c6** будет больше на 10мм, при значении **d=10** – меньше на 10мм.

Общие обозначения, встречающиеся в форматах:

cn - устанавливает название окружности индекса n.

Все индексы: **n, m,r,q** и **p** являются целыми порядковым числами, заключёнными между **1** и максимальным числом определенным при конфигурации системы ЧПУ.

I..J.. - координаты центра окружности;

r.. - радиус окружности (положительный (знак не указывается) при движении против часовой стрелки и отрицательный со знаком минус - по часовой стрелке);

[-] lm lp - предопределённые элементы прямой с индексом **m** и **p;**

[-]lp - может принять противоположное направление при использовании знака «-» минус;

pm pq pr - предопределённые элементы точки индекса **m,q,r;**

[-]cm[-]cp - предопределённые элементы окружности с индексом **mp;** могут принять противоположное направление при использовании знака «-» минус;

1.6.2 Формат косвенного программирования окружности

Формат:

- окружность с данным радиусом и касательная к двум предопределённым прямым (рис. 1.41):

cn=[-]lm,lp,r..

c3=l1,l2,r-15

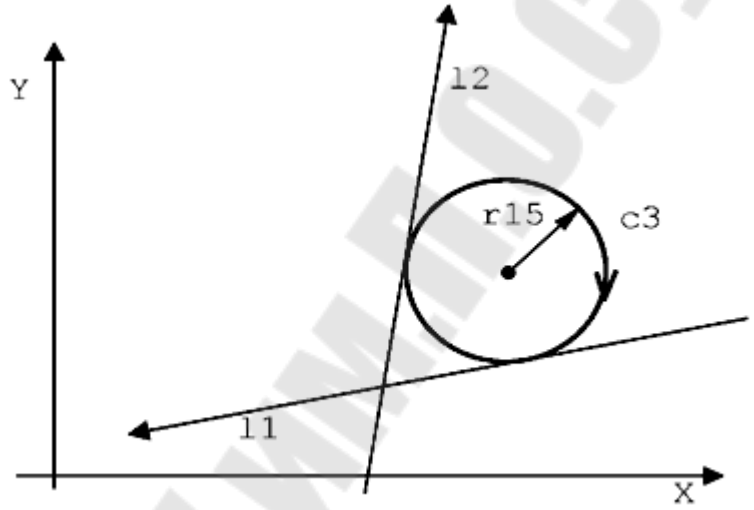

Рис.1.41.Определение окружности с данным радиусом и касательная к двум предопределённым прямым.

Формат:

- окружность с данным радиусом и касательная к (предопределёнными) прямой и окружности (рис.1.42;1.43;1.44):

cn=[-]lm,[-]cp,r..

cn=[-]cp,[-]lm,r..

c3=l1,-c2,r8
c4=-c2,l1,r8

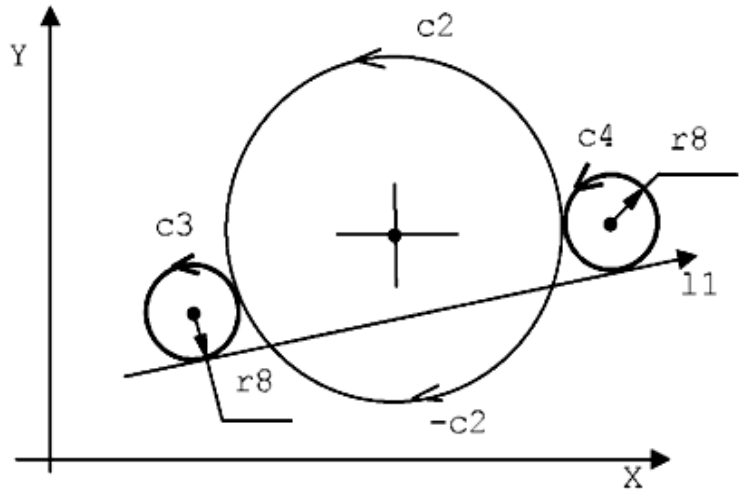

Рис.1.42 Определение окружности с данным радиусом и касательной к (предопределёнными) прямой и окружности

c9=-c2,l1,-r8
c10=l1,-c2,r-8

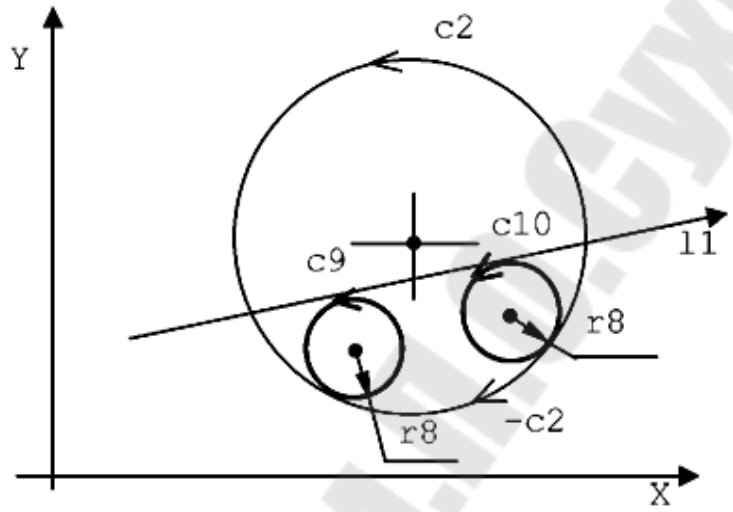

Рис. 1.43. Определение окружности с данным радиусом и касательной к (предопределёнными) прямой и окружности

c4=-l2,c1,r-40
c5=c1,-l2,r-40

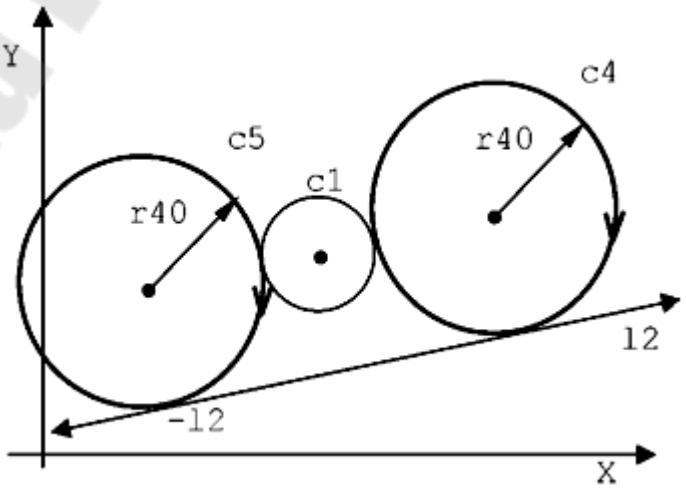

Рис. 1.44.Определение окружности с данным радиусом, и касательной к (предопределённой) прямой и окружности

Формат:

- окружности с данным радиусом и касательной к двум (предопределенным) окружностям (рис.1.45)

cn=pm,[-]lp,r..

cn=[-]lp,pm,r..

c3=p1,-l1,r25
c4=-l1,p1,r25

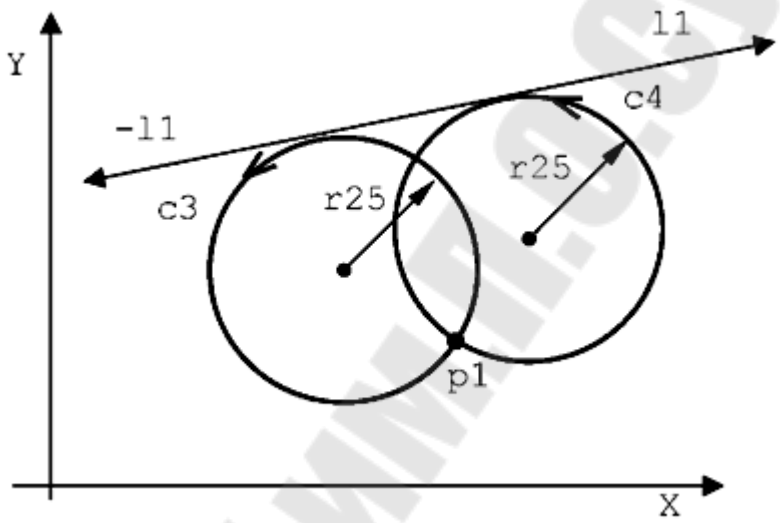

Рис.1.45 Определение окружности с данным радиусом и касательной к двум (предопределенным) окружностям

Формат:

- окружность с данным радиусом и касательная к двум предопределённым окружностям (рис.1.46; 1.47).

cn=[-]cm,[-]cp,r..

c5=c1,c2,r-8
c6=c2,c1,r-8

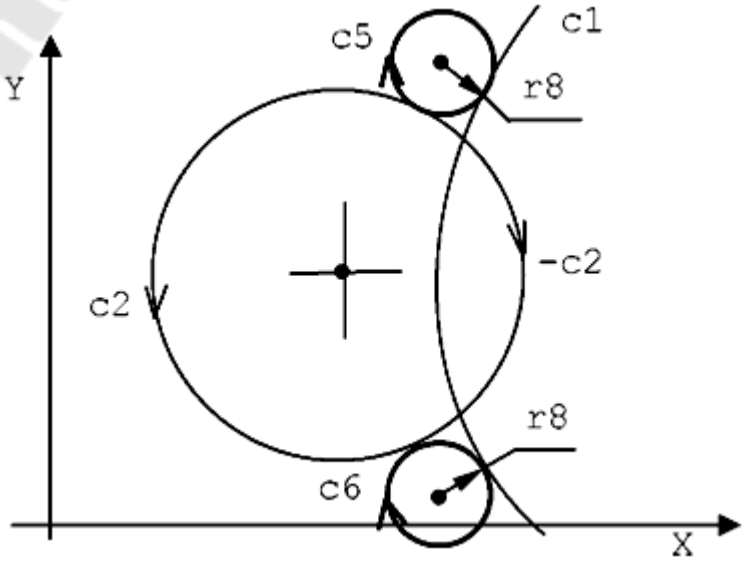

Рис.1.46 Определение окружности с данным радиусом и касательная к двум предопределённым окружностям

c9=-c2,c1,r8
c10=c1,-c2,r-8

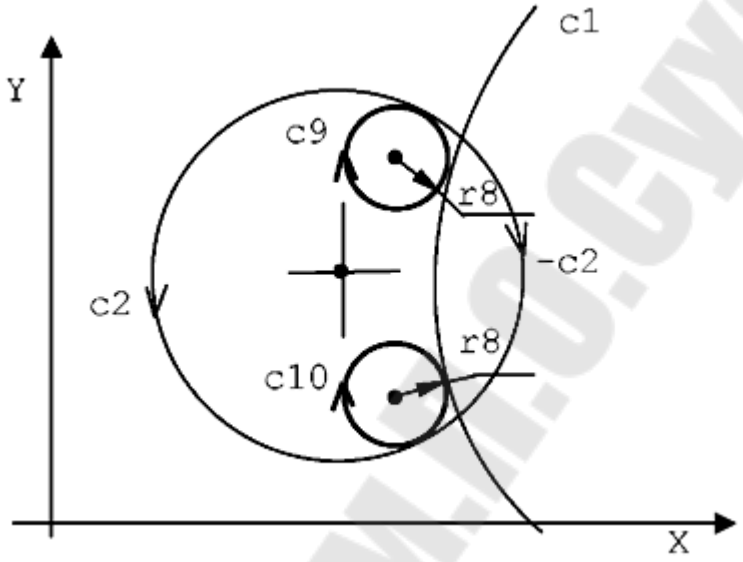

Рис. 1.47.Определение окружности с данным радиусом и касательной к двум предопределённым окружностям.

Формат:

- окружность с данным радиусом, проходящая через одну предопределённую точку, касательная к предопределённой окружности (рис.1.48):

cn=pm,[-]cp,r..
cn=[-]cp,pm,r..

c2=c1,p1,r60
c3=p1,c1,r60

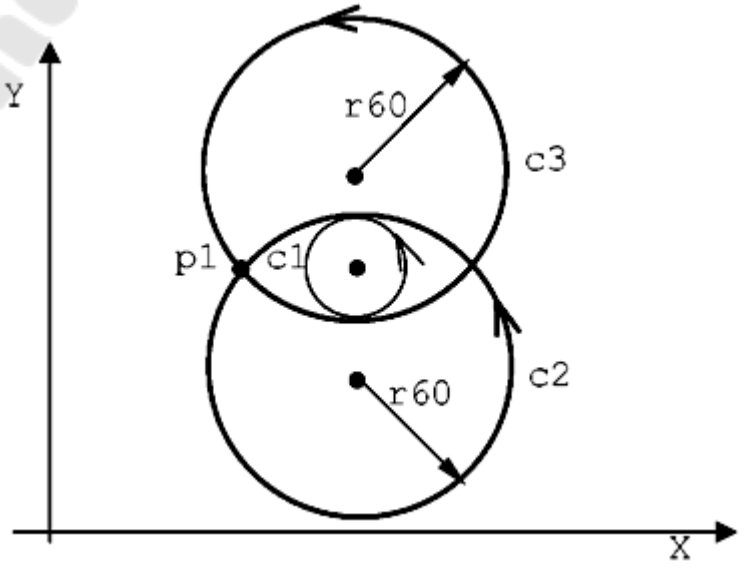

Рис. 1.48. Определение окружности с данным радиусом, проходящей через одну предопределённую точку, касательной к предопределённой окружности.

Формат:
- окружность с данным радиусом, проходящая через две предопределённые точки (рис.1.49):
cn=pm,pq,r..

c1=p1,p2,r20
c2=p2,p1,r20

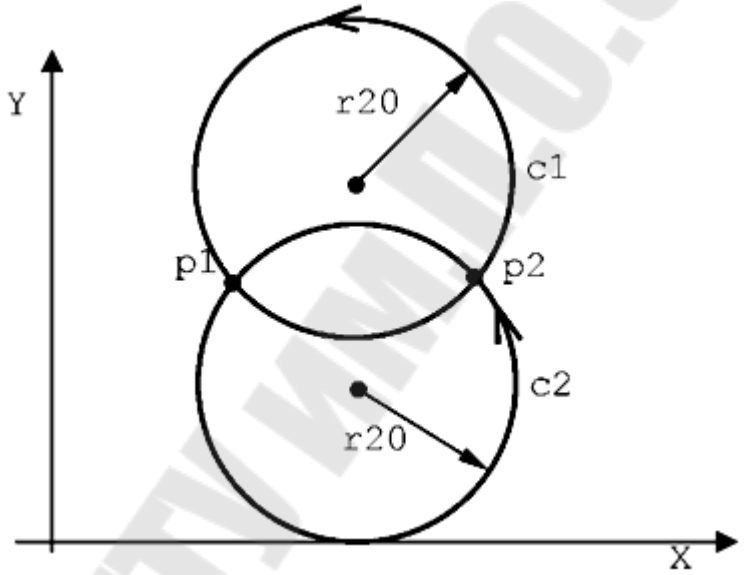

Рис.1.49. окружность с данным радиусом, проходящая через две предопределённые точки.

Формат:
- окружность с центром в предопределенной точке и касательная к предопределённой прямой (рис. 1.50):
cn=pm,[-]lp

c3=p1,l1

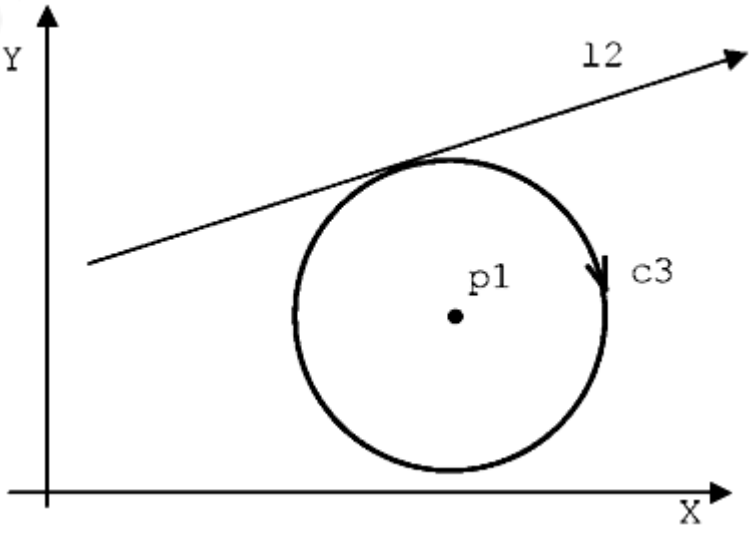

Рис. 1.50. Оп-

Определение окружности с центром в предопределенной точке и касательной к предопределённой прямой.

Формат:

- окружность с центром в предопределенной точке и касательная к предопределённой окружности (рис.1.51)

cn = pm,[-]cp[,s2];

c2=p1,c1
c3=p1,c1,s2

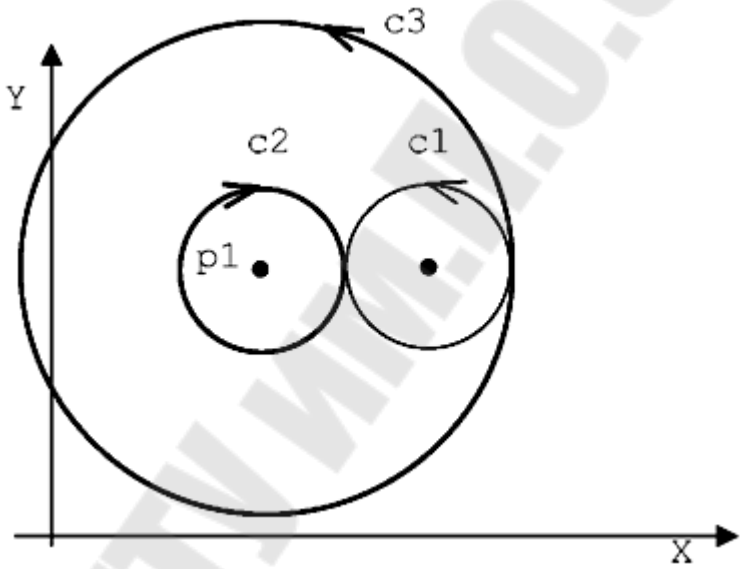

Рис. 1.51.Определение окружности с центром в предопределенной точке и касательная к предопределённой окружности.

Примечание:

При применении дискриминатора **s2** образуется окружность **c3** с большим диаметром, с направлением совпадающим с окружность **c1**.

Формат:

- окружность, проходящая через три точки (рис.1.52):

cn=pm,pq,pr

c1=p1,p2,p3
c2=p3,p2,p1

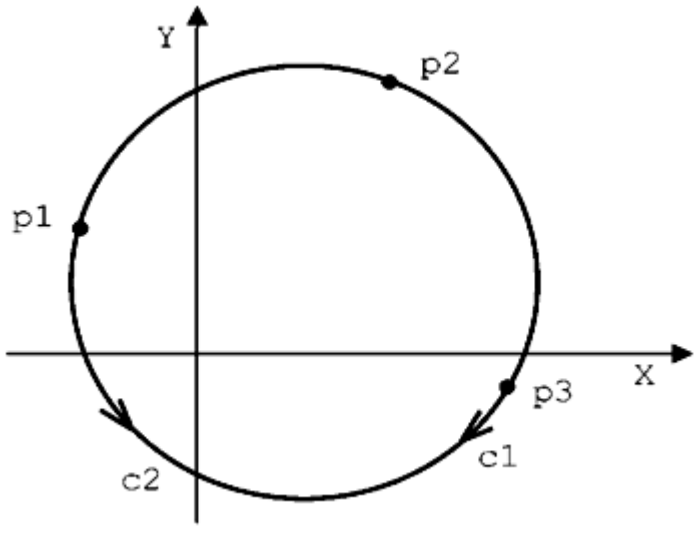

Рис.1.52.Определение окружности, проходящей через три точки.

Формат:

- окружность с данным радиусом и с центром в предопределенной точке (рис.1.53):

cn=pm,r..

c1=p1,r-40

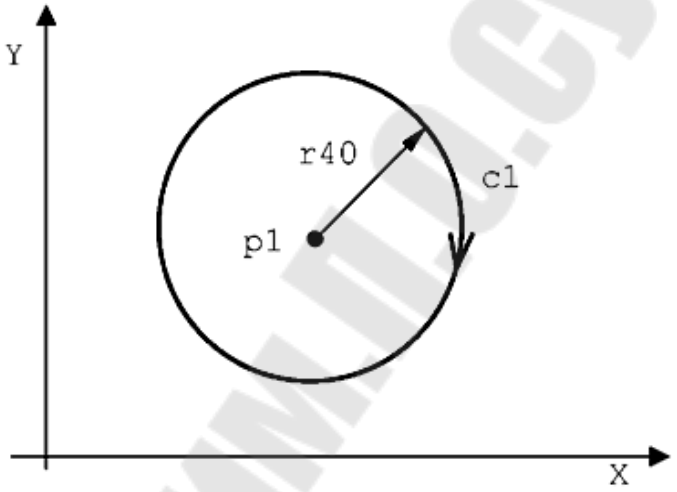

Рис.1.53.Определение окружности с данным радиусом и с центром в предопределенной точке

Формат:

- окружность концентрическая к предопределённой окружности и отдалённая от неё на данную величину (рис.1.54):

cn=[-]cm,d..

c6=c5,d-10

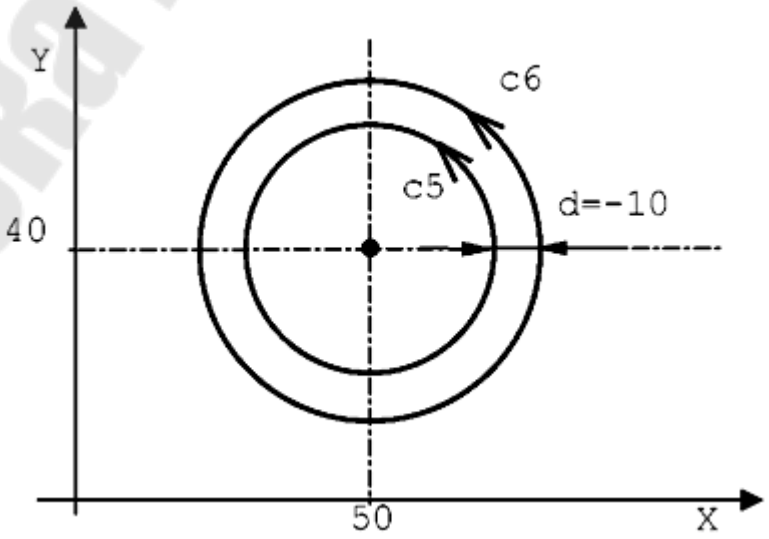

Рис..1.54.Определение концентрической окружности к предопределённой окружности и отдалённая от неё на данную величину.

Общие обозначения, встречающиеся в форматах:

ln - определяет название прямой с индексом **n** (**n** – порядковое число, заключённое между 1 и максимальным числом определенным при конфигурации системы ЧПУ);

cn- устанавливает название окружности индекса **n** (**n** – порядковое число, заключённое между 1 и максимальным числом определенным при конфигурации системы ЧПУ);

I..J.. - координаты центра окружности;

r.. - радиус окружности (положительный для направления против часовой стрелки, отрицательный для направления по часовой стрелке).

[-]lm - предопределённые элементы прямой с индексом **m,** может принять противоположное направление при использовании знака «-» минус;

[-]lp - предопределённые элементы прямой с индексом **p**, может принять противоположное направление при использовании знака «-» минус;

pm pq pr - предопределённые элементы точки индекса **m, q,r;**

[-]cm[-]cp - предопределенные элементы окружности с индексом **m p,** могут принять противоположное направление при использовании знака «-» минус;

[s2] - атрибут для наибольшей из двух возможных окружностей;

d.. - расстояние между двумя окружностями: - положительное, если глядя в направлении движения по предопределенной окружности т.е. **cm, cn** находится слева от неё, и отрицательное, если находится справа.

1.7.Определение профиля

Под профилем подразумевается последовательность геометрических элементов, накопленных в памяти системы до начала обработки. Профиль может быть ***открытым и закрытым.***

1.7.1.Начало и конец профиля

Профиль, запрограммированный в геометрии GTL, определяется через функции **G21** и **G20:**

G21 - устанавливает начало профиля;

G20 - устанавливает конец профиля.

1.7.2.Открытый профиль

Если профиль открытый, он должен начинаться с точки **(pn)** и заканчиваться точкой, отличной от первой. Компенсация радиуса ин-

струмента **G41,G42**, действует перпендикулярно к первому элементу на точке начала профиля и перпендикулярно к последнему элементу на точке конца профиля. Компенсация радиуса должна быть открыта на первой точке профиля программированием в кадре с функцией **G21** дополнительно функциями **G41,G42**, и закрыта на последней точке в кадре с функцией **G20** функцией **G40**, как показано на рис. 1.55.

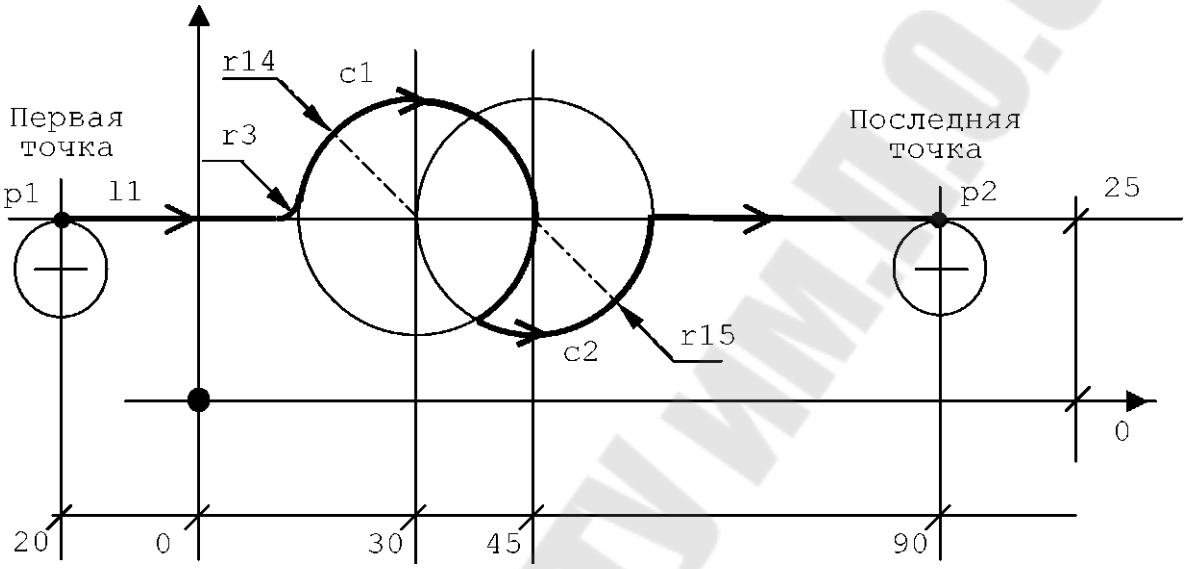

Рис.1.55. Открытый профиль

Предопределение элементов профиля явным образом.

……

l1=X0Y25,a0

p1=X-20Y25

p2=X90Y25

c1=I30J25r-14

c2=I45J25r15

……

G21G42p1 - первая точка

Z-10

l1

r3

c1s2

c2s2

l1

G20G40p2 - последняя точка

……...

Компенсация радиуса **G42** должна быть открыта на первой точке профиля и закрыта **G40** на последней. Компенсация радиуса отменяется в кадре следующим за кадром, где запрограммирована функция **G40,** при движении осей в плоскости профиля.

1.7.3.Закрытый профиль.

Если профиль закрытый, сначала следует запрограммировать последний элемент **l5** (рис.1.56), а затем после последнего элемента, вызвать первый элемент профиля **c1**.

Первая точка скорректированного профиля, это пересечение первого **c1** и последнего **l5** смещённых элементов равна ***последней точке***.

Компенсация радиуса (**G41** или **G42**) должна быть открыта в начале профиля в кадре вызова последнего элемента **l5** программированием функциями **G21** и закрыта в конце профиля в кадре вызова первого элемента **c1** с функциями **G20 G40**.

Если первый и/или второй элементы являются окружностями, возможны два пересечения. Если не даётся никакой дополнительной информации, система выбирает первое пересечение. В случае если необходимо выбрать второе пересечение, следует программировать дискриминатор **s2.** Дискриминатор **s2** программируется ***три раза: в кадре вызова последнего элемента в начале профиля и на последнем элементе в конце профиля*** (рис.1.53).

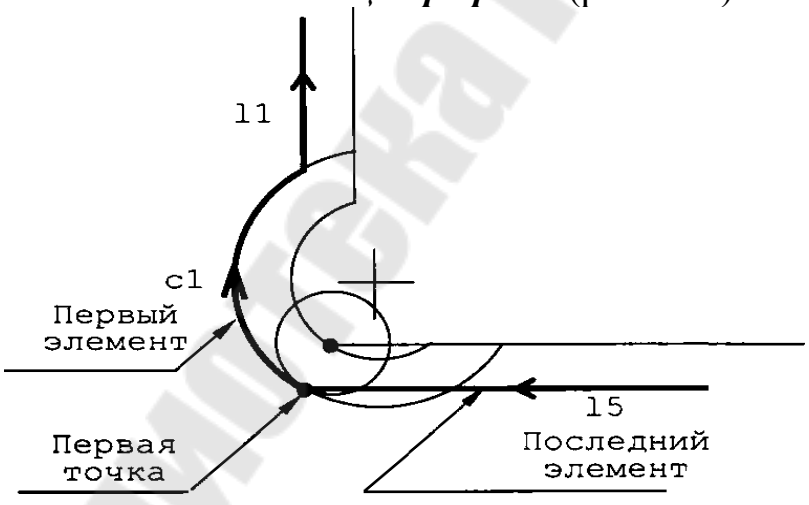

Рис.1.56. Закрытый профиль.

c1=I..J..r..

……

l1=X..Y..,a90

……

l5=X..Y..,a180

……...

G21G4215s2 - последний элемент

c1 s2 - первый элемент

l1

……

l5s2 - последний элемент

G20 G40 c1 - первый элемент

…….

Компенсация радиуса инструмента (G**41** или **G42**) должна быть открыта в начале профиля в кадре вызова последнего элемента и закрыта **G40** в конце профиля в кадре вызова первого элемента. Компенсация радиуса инструмента отменяется в первом кадре движения осей в плоскости профиля, следующего за функцией **G40** (рис.1.57).

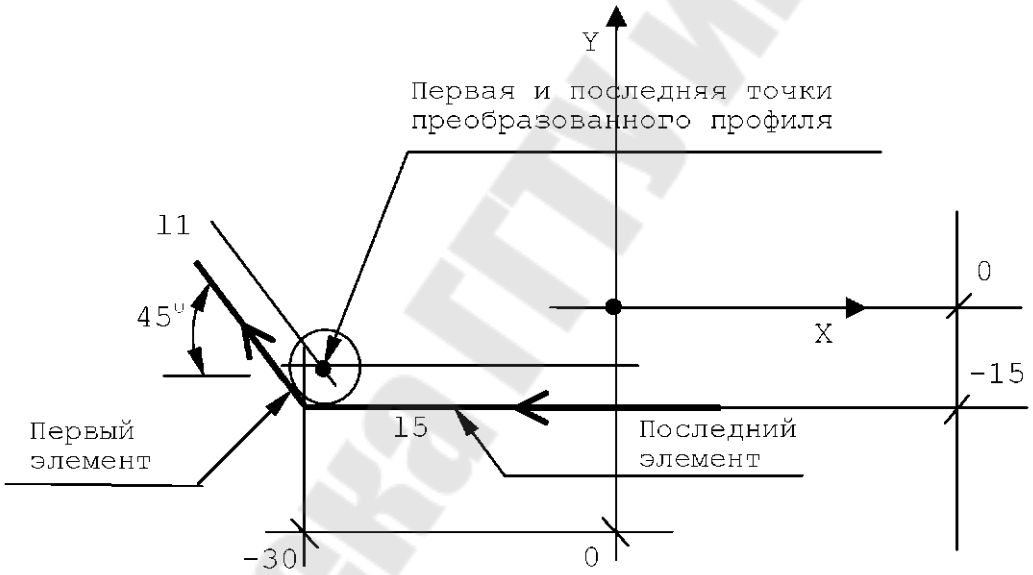

Рис.1.57. Закрытый профиль.

…….

l5=X0Y-15,a180 – последний элемент

…….

l1=X-30Y-15,a135 – первый элемент

…….

G21G42l5

Z-10

l1

……
l5
G20G40l1

1.7.4.Движение осей шпинделя

В любой точке профиля представляется возможным двигать оси, не участвующие в контурной обработке, даже на первой точке, например, для входа в деталь. В случае открытых профилей движение на первой точке программируется после программирования первой точки (рис.1.55).

Открытый профиль.

……
G21 G42 p1 – первый элемент.
Z-10
l1

В случае закрытых профилей оно должно быть запрограммировано между определением последнего элемента профиля и первым элементом (рис.1.57).

Закрытый профиль.

…….....
G21G42l5 – последний элемент.
Z-10
l1 – первый элемент.

1.7.5.Соединение геометрических элементов.

Геометрические элементы профиля могут быть связаны между собой за счёт тангенциального сопряжения, пересечения или присутствия автоматического соединения или фаски.

В случае пересечения двух прямых, возможно только одно решение. В случае пересечения прямая-окружность или окружность-окружность всегда возможны два решения. Система автоматически выбирает первое. Если необходимо получить второе, следует запрограммировать дискриминатор **s2** после определения первого элемента. Примеры пересечения прямая-окружность приведены на рис.1.58.

В случае пересечения прямой с окружностью первое и второе пересечения определяются направлением движения прямой линии. В

случае пересечения окружности с окружностью (рис.1.59) первым пересечением является то, что слева от прямой, соединяющей центр первой окружности с центром второй, а второе **s2** пересечение то, что справа от той же прямой линии.

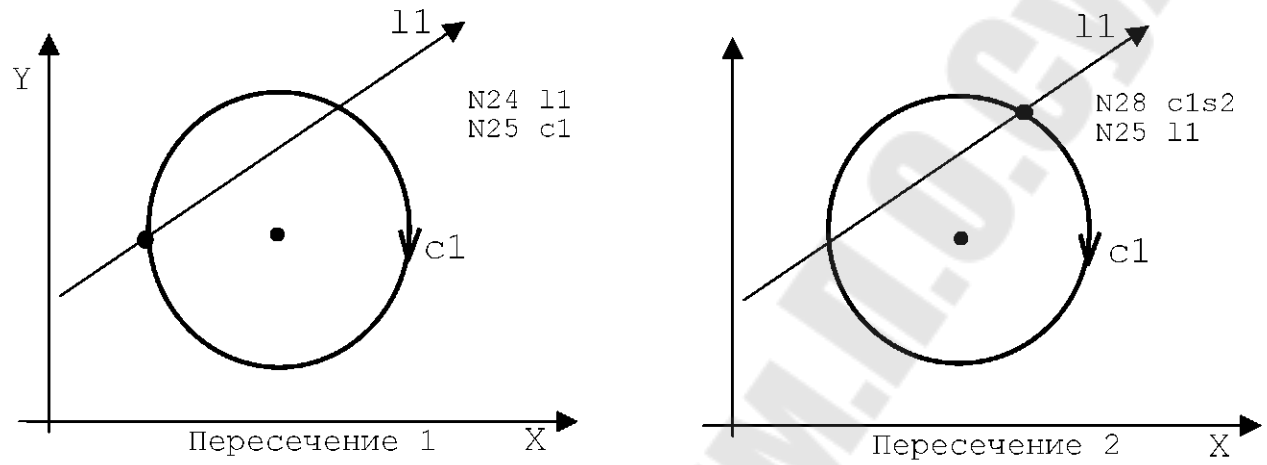

Рис.1.58.Определени точек пересечения прямой с окружностью.

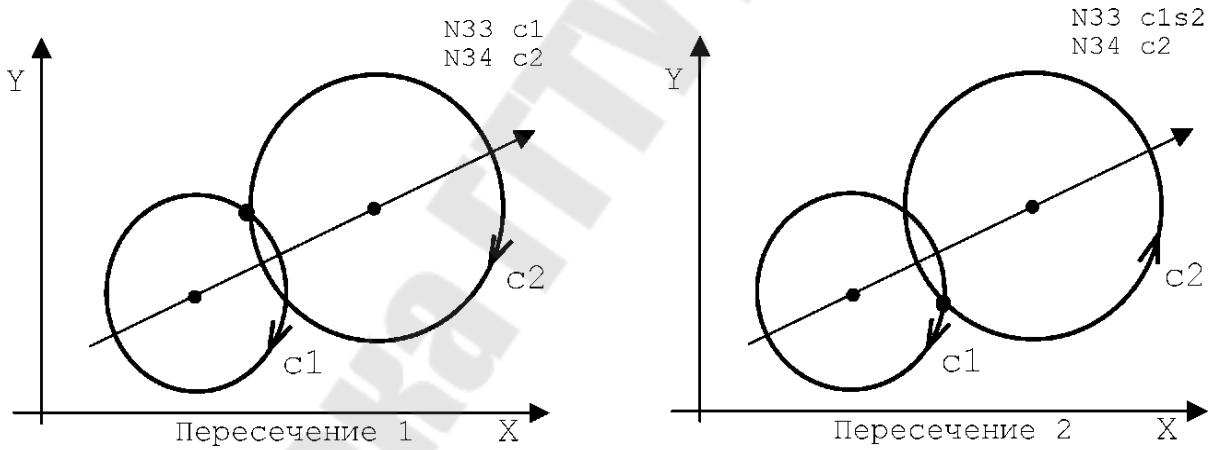

Рис.1.59. Определение точек пересечения двух окружностей.

1.7.6.Программирование соединения между элементами при помощи автоматического радиуса r.

Если элементы пересекаются, можно определить соединение между ними (прямые линии или окружности), программируя значение радиуса с указанием его численного значения: ***положительное в направлении против часовой стрелки, отрицательное со знаком «-» минус в направлении по часовой стрелке.*** Пример приведён на рис.1.60.

Соединение **r** не может быть запрограммировано в кадре, следующем сразу же за кадром с **G21**, или же в кадре, предшествующем кадру с **G20**, т. е. профиль не может начинаться и заканчиваться **r** соединением.

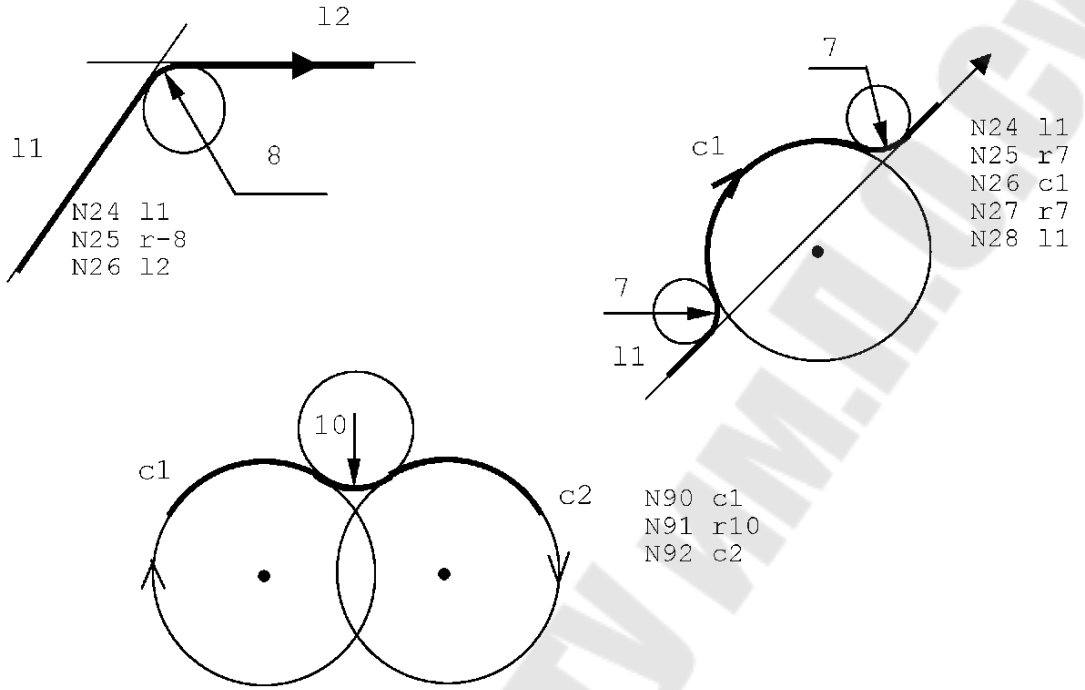

Рис.1.60.Программирование **r** соединения.

В случае активизации компенсации радиуса инструмент размещается на пересечении двух геометрических элементов, соединенных радиусом инструмента. Для сокращения пути перемещения инструмента необходимо ввести нулевой радиус **r0** между двумя элементами, в соответствии с рис. 1.61.

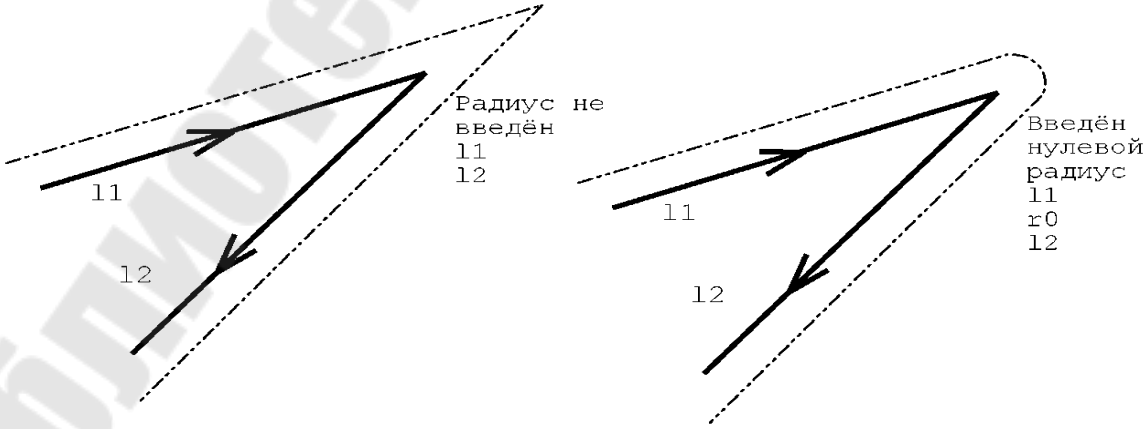

Рис.1.61. Программирование нулевого радиуса **r0**.

1.7.7.Программирование скосов.

Скосы между прямолинейными элементами можно определить, программируя значение скоса без знака, рассматриваемое как расстояние от точки пересечения. Пример приведён на рис.1.62.

Скос не может быть запрограммирован в кадре, который непосредственно следует за кадром G21 или предшествует G20 (т. е. профиль не может начинаться или заканчиваться скосом).

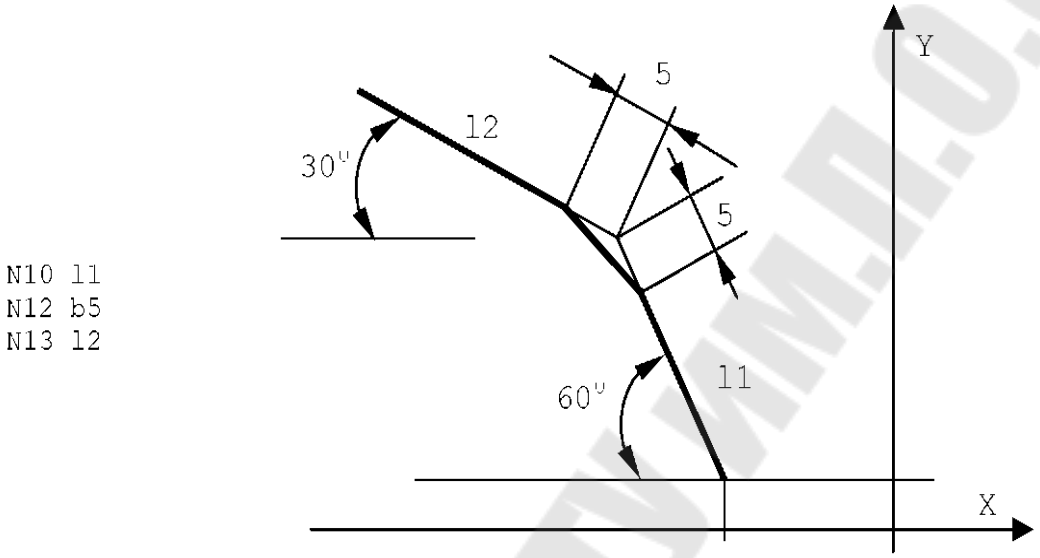

Рис.1.62.Программирование скосов.

В геометрическом программировании GTL перемещения всегда осуществляются с рабочей подачей, для программирования быстрого перемещения необходимо программировать скорость рабочей подачи **F** со значением быстрого хода с функцией **G01**.

Если плоскость интерполяции не является той, которая образована осями X и Y, следует вначале определить плоскость интерполяции с помощью трехбуквенного кода DPI, например, (DPI, Z, X), а затем определять элементы профиля.

1.8.Примеры программирования на языке GTL.

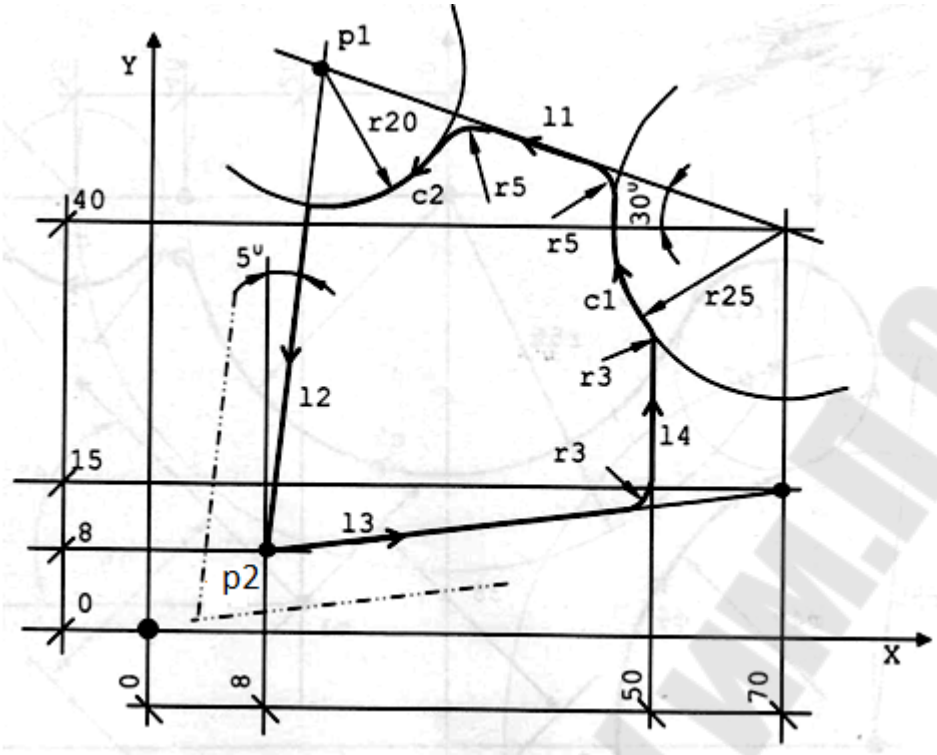

Рис.1.63.Программирование профиля 1 на языке GTL.

```
;PROF1
N5(DIS,"PROFIL 1")
N10(UAO,1)
N20T1.1M06
N30(UCG,1,X0X100,Y0Y100,Z0)
N40l1=X70Y40,al50
N5012=X8Y8,a-95
N60p1=11,12
N70p2=X8Y8
N8013=X8Y8,X70Y15
N9014=X50Y0,a90
N100cl=I70J40r-25
N110c2=p1,r-20
N120G94G97S800F250M3M8
N130GX0Y0
N140Z-10
N150G21G4212
N16013
N170r3
N18014
N190r3
N200cl
N210r5
N22011
N230r5
N240c2s2
N25012
N260G20G4013
N270G0Z2
N280G0X0Y0M30
```

Комментарии к программе PROF1:

1.В кадре N240 применен дискриминатор **s2** в связи с тем, что прямая **l2**, имеет второе пересечение с окружностью **c2.**

2.Профиль запрограммирован. как закрытый, см. кадры N150 и N260.Возможен другой вариант их программирования: **N150G21G42p2**; **N260G20G40p2**. Для этого необходимо знать точно заданные на профиле координаты точки **p2** и ее предопределить, см. кадр N70.

3.Программа должна быть написана одним столбцом в текстовом редакторе «Блокнот», шрифт «По умолчанию».

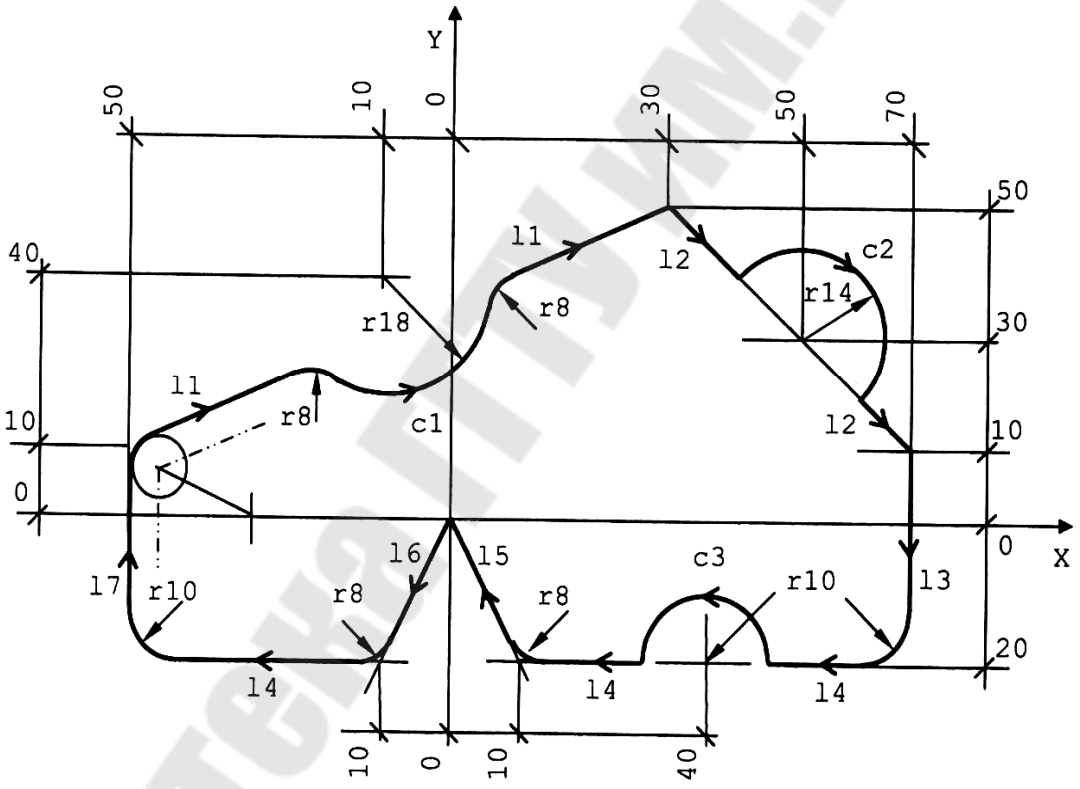

Рис 1.64 Программирование профиля 2 на языке GTL.

```
;PROF2
N5(DIS,"PROFIL 2")
N10(UAO,1)
N20T1.1M06
N30(UCG,1,X-100X100,Y-50Y80,Z0)
N4011=X-50Y10,X30Y50
N5012=X30Y50,X70Y10
N6013=X70Y0,a-90
N7014=X0Y-20,a180
N8015=X10Y-20,X0Y0
N9016=X0Y0,X-10Y-20
N10017=X-50Y0a90
N110cl=I-10J40rl8
N120c2=I50J30r-14
```

N130c3=I40J-20r10
N140G94G97S800F250M3M8
N150G0X-30Y0
N160Z-10
N170G21G4217
N18011
N190r-8
N200r-8
N21011
N22012
N230c2s2
N24012
N250cl
N26013
N270r-10
N28014
N290c3s2
N30014
N310r-8
N32015
N33016
N340r-8
N350r-8
N36014
N37014
N380r-10
N39017
N400G20G4011
N410G0Z0

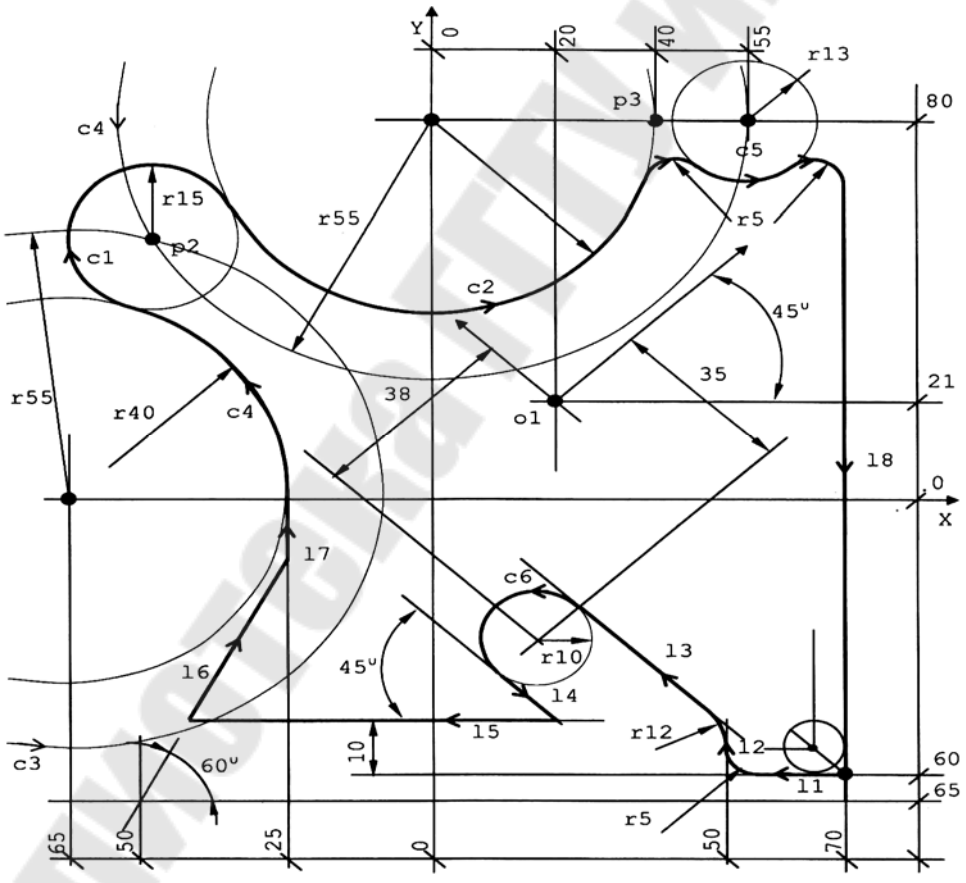

Рис 1.65 Программирование профиля 3 на языке GTL.

```
;PROF3
N5(DIS,"PROFIL 3")
N10(UAO,1)
N20T1.1M06
N30(UCG,1,X0X100,Y0Y100,Z0)
N40ol=X20Y21a45
N501l=X0Y-60a180
N60l2=X50Y0,a90
N70c6=olI-38J-35rl0
N8013=c6,a135
N90l4=c6,a-45
N100l5=X0Y-50,a180
N110l6=X-50Y-65,a60
N120l7=X-25Y0,a90
N130c3=I-65J0r55
N140c4=I0J80r55
N150p2=c3,c4
N160cl=p2,r-15
N170p3=X40Y80
N180c2=cl,p3,r40
N190c5=I55J80rl3
N200l8=X70Y0,a-90
N210G94G97S800F250M3M8
N220G21G42l8
N230Z-10
N240l1
N250r-5
N260l2
N270r12
N28013
N290c6
N300l4
N310l5
N320l6
N330l7
N340r40
N350cl
N360c2
N370r-5
N380c5
N390r-5
N400l8
N410G20G40l1
N420GZ
N430X..Y..M30
```

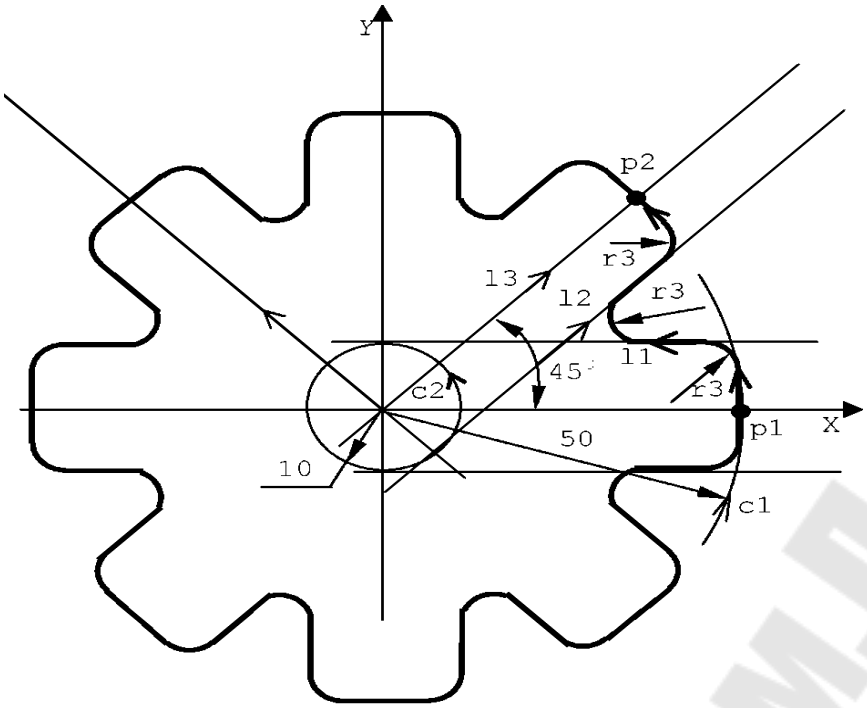

Рис 1.66.Программирование профиля ZVEZDA с поворотом осей.

```
;ZVEZDA
N10(UAO,1)
N20T1.1M06
N30(UCG,1,X80X80,Y80Y80,Z0)
N40UOV=2
N50p1=X50 Y0
N60c1=I0J0r50
N70c2=I0J0r10
N80l1=c2,a180
N90l3=X0Y0,a45
N100l2=c2,a4
N110p2=l3,c1,s2
N120G94G97S800F250M3M8
N130X60Y0
N140Z-20
"START"N13E25=0
N150(RPT,8)
N160(URT,E25)
N170G21G42p1
N180c1
N190r3
N200l1
N210r-3
N220l2
N230r3
N240c1
N250G20G40p2
N260E25=E25+45
N270(ERP)
N280(URT,0)
"END"N28
N290UOV=0
N300(EPP,START,END)
N310G0Z20
N320X0Y0M30
```

Литература

1. Гжиров Р. И., Серебреницкий П. П. Программирование обработки на станках с ЧПУ/ Р.И.Гжиров, П.П.Серебреницкий Справочник.– Л.: Машиностроение, 1990.–558 с.
2. Схиртладзе А. Г., Технологическое оборудование машиностроительных производств/ А. Г., Схиртладзе В. Ю Новиков, Учеб. пособие для машиностроит. спец. вузов / Под ред. Ю.М. Соломенцева,2е изд. перераб. и доп.–М.: Высш. шк., 2001–407с.
3. Фельдштейн. Е.Э Обработка деталей на станках с ЧПУ/ Е.Э.Фельдштейн, М.А.Корниевич.- Минск, «Новое знание», 2005г, - 287с.
4. Руководство оператора (УЧПУ NC-201, NC-201M, NC-202), ООО «Балт-Систем», Санкт-Петербург, 2008г, www.bsystem.ru.
5. Руководство программиста (УЧПУ NC-110, NC-201, NC-201M, NC-202, NC-210, NC-220, NC-230), ООО «Балт-Систем», Санкт-Петербург, 2008г, www.bsystem.ru.
6. Пучков А.А. Устройство управления РБ242Б роботами М10П.62.01/ А.А.Пучков. Учебное пособие. – Мн.: БПИ, 1990.– 62с.
7. Пучков А.А. Практикум по токарным робото–технологическим комплексам/ А.А.Пучков Учеб. пособие для вузов. – Мн.: Выш. шк., 1992. – 168с.

Старовойтов Николай Андреевич
Дмитриченко Евгений Эдуардович

ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ ВЫСОКОГО УРОВНЯ GTL ПРИ РАЗРАБОТКЕ УПРАВЛЯЮЩИХ ПРОГРАММ ДЛЯ СВЕРЛИЛЬНО-ФРЕЗЕРНО-РАСТОЧНЫХ СТАНКОВ С ЧПУ

Практикум
по дисциплине «Технология обработки на станках с ЧПУ» для студентов специальности 1-36 01 01 «Технология машиностроения» дневной и заочной форм обучения

Подписано к размещению в электронную библиотеку
ГГТУ им. П. О. Сухого в качестве электронного
учебно-методического документа 16.05.18.

Рег. № 80Е.
http://www.gstu.by