GERİ ÇEKME (PULLBACK) CAT^1-LİE CEBİRLERİ

Ezgi YAVUZ

Yüksek Lisans Tezi

Matematik Anabilim Dalı

Mart – 2008

GERİ ÇEKME (PULLBACK) CAT^1-LİE CEBİRLERİ

Ezgi Yavuz

Dumlupınar Üniversitesi
Fen Bilimleri Enstitüsü
Lisansüstü Yönetmeliği Uyarınca
Matematik Anabilim Dalında
YÜKSEK LİSANS TEZİ
Olarak Hazırlanmıştır.

Danışman : Doç. Dr. Murat ALP

Mart - 2008

KABUL VE ONAY SAYFASI

Ezgi YAVUZ' un YÜKSEK LİSANS tezi olarak hazırladığı GERİ ÇEKME (PULLBACK) CAT^1-LİE CEBİRLERİ başlıklı bu çalışma, jürimizce lisansüstü yönetmeliğin ilgili maddeleri uyarınca değerlendirilerek kabul edilmiştir.

…/03/2008

Üye : Prof. Dr. Zekeriya ARVASİ

Üye : Doç. Dr. Murat ALP

Üye : Yrd. Doç. Dr. Erdal ULUALAN

Fen Bilimleri Enstitüsünün Yönetim Kurulu'nun …../…../….. gün ve ….. sayılı kararıyla onaylanmıştır.

Prof. Dr. M. Sabri ÖZYURT
Fen Bilimleri Enstitüsü Müdürü

GERİ ÇEKME (PULLBACK) CAT^1-LİE CEBİRLERİ

Ezgi YAVUZ

Matematik Bölümü, Yüksek Lisans Tezi, 2008
Tez Danışmanı: Doç. Dr.Murat ALP

ÖZET

Bu tez dört bölümden oluşmaktadır. İlk bölüm teze kısa bir giriştir. Bölüm iki gruplar üzerinde tanımlanmış çaprazlanmış modüller ve GAP uygulamaları ile birlikte bu kategoriye denk olan cat^1-gruplar kategorisini içermektedir. Ayrıca bu bölüm, geri çekme çaprazlanmış modül ve geri çekme cat^1-grup kavramlarını da içermektedir. Bölüm üç Lie cebirleri, Lie braketi ve uygulamaları ile birlikte Lie cebirlerinin GAP uygulamalarını ve bu cebirler üzerinde tanımlanmış Lie çaprazlanmış modül ve cat^1-Lie cebir kavramlarını da içermektedir. Bölüm dört geri çekme Lie çaprazlanmış modül ve denk kategorisi geri çekme cat^1-Lie cebir kategorisini içermektedir.

Anahtar Kelimeler : Cat^1-gruplar, Çaprazlanmış Modüller, Lie cebir, Geri çekme Çaprazlanmış Modüller, Geri çekme cat^1-gruplar, Geri çekme cat^1-Lie cebir, GAP.

PULLBACK CAT^1-LİE ALGEBRAS

Ezgi YAVUZ

Department of Mathematics, MsC.Thesis, 2008
Thesis Supervisor:Assoc.Prof. Murat ALP

SUMMARY

This thesis consists of four chapters. First chapter is a short introduction to the thesis. Chapter two includes crossed modules, crossed modules morphisms, examples of crossed modules, application method of crossed modules to the computer using the computer programme GAP (Group Algorithm and Programming) and application examples. This chapter also includes pullback crossed modules and pullback cat^1-groups. Chapter 3 includes the notation of Lie algebra, Lie algebra bracket and GAP applications of Lie algebra. Crossed modules of Lie algebras and cat^1-Lie algebra is also presented in this chapter. Chapter 4 contains the notion of pullback crossed module of Lie algebras and pullback cat^1-Lie algebras.

Key Words : Cat^1-groups, Crossed modules, Lie algebra, Pullback crossed modules, Pullback cat^1-groups, Pullback cat^1-Lie algebra, GAP.

TEŞEKKÜR

Yüksek lisans çalışmamı yöneten ve çalışma süresince vakit ayırarak ilgi ve yardımlarını esirgemeyen değerli hocam, Sayın Doç. Dr. Murat ALP'e ayrıca, her zaman yanımda olan ve desteklerini hiçbir zaman esirgemeyen aileme, sonsuz saygı ve teşekkürlerimi sunarım.

İÇİNDEKİLER

Sayfa

ÖZET iv

SUMMARY v

SİMGELER DİZİNİ viii

1. GİRİŞ 1

2. ÇAPRAZLANMIŞ MODÜLLER 4

2.1. Çaprazlanmış Modülün Örnekleri 4
2.2. Çaprazlanmış Modül Morfizmi 15
2.3. Geri Çekme (Pullback) Çaprazlanmış Modüller 17
2.4. Cat^1-Gruplar 19
2.5. Cat^1- Grup Morfizmi 26
2.6. Geri Çekme (Pullback) Cat^1-Gruplar 29

3. LİE CEBİRLERİ ÜZERİNDE ÇAPRAZLANMIŞ MODÜLLER 32

3.1. Lie Cebirleri 32
3.2. Çaprazlanmış modüller 42
3.3. Lie cebirlerinin Çaprazlanmış Modüllerinin Standart Örnekleri 43
3.4. Cat^1-Lie Cebirleri 45

4.GERİ ÇEKME (PULLBACK) LİE CEBİR ÇAPRAZLANMIŞ MODÜL 48

4.1.Geri Çekme (Pullback) Cat^1- Lie Cebir 50

KAYNAKLAR 60

SİMGELER DİZİNİ

Simgeler	**Açıklama**
$\propto$	Yarı-direkt çarpım
XMOD	Çaprazlanmış Modül Kategorisi
X	Çaprazlanmış Modül
∂	Çaprazlanmış Modül homomorfizmi
$\iota^{*}(X)$	Geri çekme (Pullback) Çaprazlanmış Modül
∂^{*}	Geri çekme (Pullback) Çaprazlanmış Modül homomorfizmi
CAT	Cat^1-Gruplar Kategorisi
C	Cat^1-Grup
$\iota^{**}(C)$	Geri çekme (Pullback) Cat^1-Gruplar
h, t, e	Cat^1-Grubun başlangıç, bitiş ve gömme homomorfizmleri
h^{**}, t^{**}, e^{**}	Geri çekme (Pullback) Cat^1-Grubun başlangıç, bitiş ve gömme homomorfizmleri
Ker h, Ker t	Başlangıç ve bitiş homomorfizmlerinin çekirdeği
$\langle \delta, \rho \rangle$	Çaprazlanmış Modül ve Cat^1-Grup Morfizm çifti

1. GİRİŞ

Whitehead tarafından 1949 yılında tanımlanan çaprazlanmış modüller [23] bir çok matematikçi tarafından kullanılmıştır. Çaprazlanmış modüller özellikle K-teoride, kategori teoride, gruplarda ve cebirler üzerinde çeşitli uygulama alanları bulmuştur. Bu uygulama alanları matematiksel bir çok sonuç vermekle birlikte son zamanlarda bilgisayar uygulamalarına da konu olmuştur. Bu uygulamaların başında çaprazlanmış modüller kategorisinin hesaplanabilirliği gelmektedir bir diğer kategoriksel uygulaması ise geri çekme (Pullback) Çaprazlanmış Modüllerdir [10].

Whitehead'in çaprazlanmış modül tanımlamasının en önemli uygulamalarından biri ise Loday [20] tarafından 1982'de verilmiştir. Loday cat^1-gruplar tanımını vermiş ve bu tanımlamadan oluşturduğu kategorinin çaprazlanmış modüller kategorisine denk olduğunu göstermiştir. Bu gösterimle denk olan kategoriyi Loday 1-cat-grup olarak tanımlamıştır. Daha sonra Loday ve Brown, bu tanımlanan kategoriyi cat^1-gruplar kategorisi olarak yeniden yapılandırmışlardır. Bu denk kategorilerin kullanılması sonucu elde edilen en yeni örnekler ise Alp tarafından tanımlanan Pullback cat^1-gruplar [2] ve Induced cat^1-gruplar [6] dır. Benzer biçimde, Actor ve Whitehead cat^1-gruplar [5], cat^1-grupların özel durumları [4] ve bilinen küçük dereceden grupların cat^1-gruplarının sıralanması [11] verilerek denk kategori olan cat^1-gruplar örneklendirilmiştir. Yine; çaprazlanmış modüller kategorisi, denk olan cat^1-gruplar kategorisi ve denk simplisel gruplar kategorisi için önemli uygulamalardan biri ise bu denk kategorilerin bilgisayar ortamına aktarılarak hesaplanabilir hale getirilmesidir. Bu işlem 1997 yılında Alp tarafından [1] yapılmış olup hesaplamalarda GAP [21] programı kullanılmıştır. GAP programının kullanılmasının nedenleri ve çaprazlanmış modüller kategorisi ile cat^1-grupların hesaplanmasının nedenleri aşağıdaki gibi sıralanabilir.

Çaprazlanmış modüller kategorisinin hesaplanabilir duruma getirilmesi halinde,

• Çaprazlanmış modüllerdeki örneklerin oluşturulmasını kolaylaştıracak ve bilinen teoriler arasındaki ilişkiler daha rahat kurulacaktır.

• Çaprazlanmış modüller iki boyutlu gruplar olarak düşünülebilir.

• Çaprazlanmış modüllerden bakıldığında grup teorideki bazı noktalar daha detaylı görülebilir.

• Çaprazlanmış modüller çift groupoidler için temel teşkil edecek konuma sahip olacaktır [17].

Yukarıda kısa bir tarihçesini verdiğimiz çaprazlanmış modüller kategorisi, cat^1-gruplar kategorisi ve GAP programı bu çalışmanın oluşmasına ve GAP programında henüz bulunmayan Lie cebirlerin incelenmesine neden olmuştur.

Çaprazlanmış modüllerde ele alınan gruplar yerine Lie-cebir alınabilirliğini inceleyebilmek için ve GAP uygulamasına kaynak oluşturabilmek için bu çalışma hazırlanmıştır. Gruplar üzerinde tanımlı çaprazlanmış Modüller ile Lie cebirlerinde tanımlanmış çaprazlanmış modüller arasındaki en belirgin fark; çaprazlanmış modüller için gerekli olan etki ve boundary dönüşümünün benzerlik göstermesidir. Bu tezde öncelikle çaprazlanmış modüller, geri çekme (pullback) çaprazlanmış modüller ve daha sonra da bu kategorilere denk olan cat^1-cebirler ve geri çekme (pullback) cat^1-Lie cebirler yer almıştır.

Bölüm 1, teze giriş olarak düşünülmüş olup, bu bölümde tez içerisinde ele alınan konuların kısa tarihçesi de sunulmuştur. Bölüm 2 de Lie cebirler üzerinde tanımlanan çaprazlanmış modüller ve cat^1-Lie cebirlerin daha kolay anlaşılabilmesi için gruplar üzerindeki temel konular ve bu konuların örnekleri yer almaktadır. Daha sonra bu kategorilerin GAP aracılığıyla hesaplanmasını veren GAP program uygulamaları da verilmektedir. Ayrıca bu bölümde, GAP programının çalışma mekanizması olan kayıt alanları çaprazlanmış modül, çaprazlanmış modül morfizmleri için oluşturulmuş ve burada sunulmuştur. Yine çaprazlanmış modül ve çaprazlanmış modül morfizmlerinin hesaplanması için geliştirilen fonksiyonların bilgisayara uygulama algoritmalarını da bu bölüm içermektedir. Yine Lie cebirleri üzerindeki çalışmalarımıza temel teşkil edecek olan Brown Higgins tarafından gruplar için tanımlanan pullback çaprazlanmış modüllerde verilmiştir. Benzer şekilde, Loday'in cat^1-grup tanımlaması [20] ve bilgisayar uygulamaları esnasındaki zorluklar nedeniyle yeniden düzenlenen cat^1-grup tanımlaması [2] ve aksiyomları da sunuldu. Loday'in cat^1-gruplar ve çaprazlanmış modüllerin denkliğini gösteren ispat tekniği de bu bölümde verilmiştir. Cat^1-gruplar ve cat^1-grup morfizmlerinin hesaplanması için geliştirilen kayıt alanları, bilgisayar fonksiyonları ve fonksiyonların çalıştırılmasına ilişkin örneklerle bu bölümde sunulmuştur. Ayrıca, kategoriler arasında denklik kullanılarak Alp tarafından tanımlanan pullback cat^1-grup tanımlaması da verilmiştir.

Bölüm 3 te Lie cebirleri ve Lie paratezleri hakkında bazı uygulamalar verildikten sonra Lie cebir örnekleri verilmiştir. Lie cebirler üzerinde tanımlanan çaprazlanmış modül, örnekleri ve morfizmleri teorik olarak sunulmuştur. Bu kategoriye denk olan cat^1-Lie cebirler kavramı ve çaprazlanmış modüller kategorisine denk olduğu bu bölümde sunulmuştur. Daha sonra Gürmen [18] tarafından değişmeli cebirler için geri çekme geri çekme (pullback) çaprazlanmış modüller kategorisine denk olan cat^1-cebir formunda geri çekme (pullback) cat^1-Lie cebir kategorisi tanımlanarak kategoriler arasındaki denklik gösterilmiştir.

2. ÇAPRAZLANMIŞ MODÜLLER

S ve *R* iki grup olsun. Eğer *R, S* üzerinde bir etki ise *s* üzerindeki *r* etkisi s^r şeklinde tanımlanır. Eğer $R \subset S$ ise *R* kendisini etkiler denir ve

$$r_1^{r_2} = r_2^{-1} r_1 r_2 \ , \ r_1 r_2 \in R$$

şeklinde tanımlanır [22].

Tanım : *S* ve *R* bir grup homomorfizmi $\partial : S \to R$ ile birlikte iki grup olsun. Eğer *R*, *S* üzerinde bir etki ise $R \to Aut(S)$ şeklinde bir homomorfizm vardır. $X = (\partial : S \to R)$ aşağıdaki aksiyomları sağlıyorsa bir çaprazlanmış modüldür denir.

CM1) $\partial(s^r) = r^{-1} \partial s r \ , \quad \forall s \in S, r \in R$

CM2) $s^{\partial s'} = s^{-1} s s' \quad , \quad \forall s, s' \in S$

Genellikle ∂ çaprazlanmış modülün Boundary Dönüşümü olarak adlandırılır.

2.1. Çaprazlanmış Modülün Standart Örnekleri

Örnek 2.1.1 . *S, R* grubunun normal alt grubu olmak üzere

$$\partial : S \to R$$
$$s \mapsto s$$

içine homomorfizmi ve

$$R \times S \to S$$

$$(r, s) \mapsto {}^r s = r s r^{-1}$$

etkisini tanımlayalım. Bu durumda ∂ homomorfizmi ${}^r s = r s r^{-1}$ şeklindeki *R* nin *S* üzerinde tanımlanan etkisi ile birlikte bir çaprazlanmış modül yapısı oluşturur. Gerçekten de;

CM1) $\partial({}^r s) = \partial(r s r^{-1})$

$$= \partial(r)\partial(s)\partial(r^{-1})$$

$$= r\partial(s) r^{-1}$$

CM2) $${}^{\partial s'}s = {}^{s'}s$$

$$= s'ss'^{-1}$$

şeklinde çaprazlanmış modül aksiyomları sağlanır.

Örnek 2.1.2 *M* bir ZG-modül olmak üzere

$$\partial = 1 : M \to G$$

$$m \mapsto 1_G$$

trivial homomorfizmi ve

$$G \times M \to M$$

$$(g, m) \mapsto m^g = gm$$

etki fonksiyonu ile birlikte bir çaprazlanmış modüldür.

CM1) $$\partial(m^g) = \partial(gm)$$

$$= 1$$

$$= g^{-1}1g$$

$$= g^{-1}\partial(m)g$$

CM2) $$(m')^{\partial m} = (m')^1$$

$$= 1m'$$

$$= m^{-1}mm'$$

$$= m^{-1}m'm \qquad \text{(M abelyan grup)}$$

Örnek 2.1.3 *K* bir grup ve

$$G = \{f_k : K \to K; f_k(k') = k^{-1}k'k\}$$

kümesi *K* nın iç otomorfizmlerinin grubu olmak üzere,

$$\partial \quad : \quad K \to G$$

$$k \mapsto f_k$$

homomorfizmi

$$G \times K \to K$$

$$(f_k, k) \to (f_{k'})^k = k^{-1}k'k$$

etki fonksiyonu ile birlikte bir çaprazlanmış modüldür.

CM1) $\partial((f_{k'})^k) = \partial(k^{-1}k'k)$

$$= \partial(k^{-1})\partial(k')\partial(k)$$

$$= \partial(k^{-1})\partial(k')f_k$$

$$= f_k^{-1}\partial(k')f_k$$

CM2) $(k')^{\partial k} = (f_k)_{k'}$

$$= k^{-1}k'k$$

Örnek 2.1.4 $1 \to N \to G \xrightarrow{\partial} Q \to 1$

grupların bir merkezcil genişlemesi olmak üzere

$$\partial \ : \ G \to Q$$

$$g \to \partial(g)$$

grup homomorfizmi,

$$s \ : \ Q \to G$$

$$q \to s(q)$$

kesit homomorfizmi yardımıyla tanımlanan

$$Q \times G \to G$$

$$(q, q) \to g^q = s(q)^{-1}gs(q)$$

etki fonksiyonu ile birlikte bir çaprazlanmış modüldür.

CM1) $\partial(g^q) = \partial(s(q)^{-1} gs(q))$

$= \partial(s(q^{-1}) gs(q))$

$= \partial(s(q)^{-1})\partial(g)\partial(s(q))$

$= q^{-1}\partial(g)q$

CM2) $(g^{'})^{\partial g} = (s(\partial g))^{-1} g^{'} s(\partial g)$

$= s((\partial g)^{-1}) g^{'} s(\partial g)$

$= s(\partial(g^{-1})) g^{'} s(\partial g)$

$= g^{-1} g^{'} g$

Ayrıca $\partial : G \to Q$ tam olduğundan $\partial(G) = Q$ dir ve

$Q/\partial(G) = Q/Q \cong \{e\}$

olduğundan (G,Q,∂) bir basit bağlı çaprazlanmış modüldür.

Yukarıda verilmiş çaprazlanmış modül ve çaprazlanmış modül örnekleri bilgisayar aracılığıyla da hesaplanmıştır. Bu hesaplamalarda grup teori açısından çok güçlü olan GAP (Group Algorithm and Programming) programı kullanılmıştır. GAP [21] programı,

- Programlama dili kolay ve anlaşılır,
- Çeşitli kullanım alanlarına sahip olup grup teorisi için güçlü,
- Problemler e-mail aracılığıyla tartışılabilir ve ücretsiz kullanılabilir,
- Program grup teoride çok büyük bir kütüphaneye sahip oluşu nedeniyle tercih edilmiştir.

GAP programının kullanılmasının kolay oluşunu ve grup teoride çok güçlü oluşunu aşağıdaki örneklerle de görebiliriz. Aşağıda kullanılan fonksiyonlar GAP programının çekirdeğinde olup grupla ilgili işlemlerde herkes tarafından kullanılmaktadır. Bunun için aşağıdaki örnekler çaprazlanmış modülün hesaplanması esnasında kullanılan GAP fonksiyonlarının daha iyi anlaşılmasını sağlayacaktır. Burada kullanılan GAP programı GAP3 versiyonudur ve uygulama örnekleri [17] den alınmıştır. Burada **Group** fonksiyonu ile üreteci verilen grup üretilmektedir. **GroupId** fonksiyonu ile elde edilen grubun GAP kütüphanesindeki kayıt bilgileri elde edilmekte olup **RecFields** fonksiyonu ile de grubun üretilmiş kayıt alanları görüntülenmektedir. **GroupHomomorphismByImages** fonksiyonu bir gruptan diğer bir gruba grup homomorfizmi üretmekte olup **IsHomomorphism** fonksiyonu da üretilen homomorfizmin, homomorfizm olup olmadığını kontrol eder.

```
gap> G:=Group((1,2,3,4,5)(6,7));
Group( (1,2,3,4,5)(6,7) )
gap> GroupId(G);
rec(
catalogue := [ 10, 1 ],
size := 10,
names := [ "10", "2x5" ],
abelianInvariants := [ 2, 5 ] )
gap> RecFields(G);
[ "isDomain", "isGroup", "identity", "generators", "operations", "isPermGroup",
"isFinite", "1", "orbit", "transversal", "stabilizer", "isParent", "stabChainOptions", "stabChain",
"size", "isAbelian", "trivialSubgroup", "derivedSubgroup", "abelianInvariants", "groupId" ]
gap> gen:=G.generators;
[ (1,2,3,4,5)(6,7) ]
gap> hom:=GroupHomomorphismByImages(G,G,gen,gen);
GroupHomomorphismByImages( Group( (1,2,3,4,5)(6,7) ),
Group( (1,2,3,4,5)(6,7) ), [ (1,2,3,4,5)(6,7) ], [ (1,2,3,4,5)(6,7) ] )
gap> RecFields(hom);
[ "isGeneralMapping", "domain", "source", "range", "generators", "genimages",
"trivimages", "preImage", "image", "operations" ]
gap> IsHomomorphism(hom);
true
```

GAP programının burada verilen küçük bir uygulamasından sonra çaprazlanmış modüllerin GAP programı aracılığıyla hesaplanması da burada sunulmuştur. Çaprazlanmış modülleri hesaplayabilmek için programın bir ortak paketi olan **XMOD** Alp ve Wensley tarafından 1997 yılında geliştirilmiştir [7]. Bu pakette çaprazlanmış modüller kategorisini ve denk kategorilerin hesaplanması için geliştirilen bilgisayar fonksiyonları yer almaktadır [17]. Bu fonksiyonların daha detaylı ve GAP4 uygulaması GAP paketinde yer alan ortak paketten elde edilebilir.

GAP programı geliştirilen fonksiyonların kayıt alanlarını kullanarak çalışır. Bu nedenle de çaprazlanmış modüllerin bilgisayara uygulanması için geliştirilen kayıt alanı aşağıdaki gibidir.

Çaprazlanmış Modül Kayıt Alanları

X.source,	∂ nın tanım grubu
X.boundary,	∂ homomorfizmi
X.range,	∂ nın değer grubu
X.aut,	S nin otomorfizmlerinin grubu
X.action,	R den **X.aut** a tanımlanan bir homomorfizm
X.isXMod,	kontrol fonksiyonu, normalde doğru
X.isDomain,	daima doğru
X.operations,	**XModOps** işlemlerinin özel bir seti
X.name,	S ve R isimlerinin birleştirilmesi

1. Bu kayıt alanlarını elde etmek için aşağıdaki bilgisayar fonksiyonları geliştirilmiş olup bu fonksiyonlar; **XMod**, **IsXMod**, **XModPrint** fonksiyonlarıdır. Ayrıca GAP programı için oluşturulmuş olan **Cat1List** listesinin kullanılmasıyla da yukarıda belirtilen çaprazlanmış modül kayıt alanları da **XModSelect** fonksiyonunun kullanılmasıyla elde edilmektedir. **XModSelect** fonksiyonu GAP programında var olan grup kütüphanesi kullanılarak oluşturulmuştur. Bu fonksiyonun kullanım formu,

gap>XModSelect (gpnum,type,norm)

biçimindedir. Bu fonksiyon girdi olarak, GAP kütüphanesindeki grup numarasını *gpnum*, çaprazlanmış modülün tipini *type* ve sadece conjugation çaprazlanmış modülün inşasında da R

değer grubunun normal alt gruplar listesindeki tanım grubunun pozisyonunu belirlemek için *norm* parametrelerini alır. Burada; *gpnum* Alp tarafından üretilen ve GAP kütüphanesinde yer alan **Cat1List** listesindeki numaralandırmalardır. *Type* ise üç aşamalıdır ki bunlar otomorfizm çaprazlanmış modül için “aut”, conjugation çaprazlanmış modül için “conj” ve *R*-çaprazlanmış modüller için “rmod” tur. Şimdi bu fonksiyonun kullanımına örnekler verelim. Bu örneklerde de kayıt alanlarında belirtilen **X.operations** çaprazlanmış modül işlem setleri ayrı ayrı görüntülenecektir [17].

2. gap> X:=XModSelect(6,2,"conj");
3. Crossed module [s3->s3]
4. gap> XModPrint(X);
5. Crossed module [s3->s3] :-
6. : Source group s3 has generators: [(1,2), (2,3)]
7. : Range group = s3 has generators: [(1,2), (2,3)]
8. : Boundary homomorphism maps source generators to: [(1,2), (2,3)]
9. : Action homomorphism maps range generators to automorphisms:
10. (1,2) --> { source gens --> [(1,2), (1,3)] }
11. (2,3) --> { source gens --> [(1,3), (2,3)] }
12. These 2 automorphisms generate the group of automorphisms.
13. gap> RecFields(X);
14. ["isDomain", "isParent", "source", "range", "boundary", "action", "aut",
15. "isXMod", "operations", "name", "isConjugationXMod"]
16. gap> RecFields(X.operations);
17. ["name", "operations", "Elements", "IsFinite", "Size", "=", "<", "in",
18. "IsSubset", "Intersection", "Union", "IsParent", "Parent", "Difference", "Representative", "Random", "Print", "Kernel", "IsAspherical", "IsSimplyConnected", "IsConjugation", "IsTrivialAction", "IsCentralExtension", "DirectProduct", "IsAutomorphismXMod", "IsZeroBoundary", "IsRModule", "InclusionMorphism",

"WhiteheadPermGroup", "Whitehead", "Norrie", "Lue", "Actor", "InnerMorphism", "Centre", "InnerActor", "AutomorphismPermGroup", "IdentityMorphism", "InnerAutomorphism"]

19. gap> X1 := XModSelect(7,1,"rmod");

20. Crossed module [c7->PermAut(c7)]

21. gap> XModPrint(X1);

22. Crossed module [c7->PermAut(c7)] :-

23. : Source group c7 has generators: [(1,2,3,4,5,6,7)]

24. : Range group = PermAut(c7) has generators: [(1,3,2,6,4,5)]

25. : Boundary homomorphism maps source generators to: [()]

26. : Action homomorphism maps range generators to automorphisms:

27. (1,3,2,6,4,5) --> { source gens --> [(1,4,7,3,6,2,5)] }

28. This automorphism generates the group of automorphisms.

29. gap> RecFields(X1);

30. ["isDomain", "isParent", "source", "boundary", "range", "aut", "action", "isXMod", "operations", "isRModuleXMod", "name"]

31. gap> X2 := XModSelect(8,4,"aut");

32. Crossed module [d8->PermAut(d8)]

33. gap> XModPrint(X2);

34. Crossed module [d8->PermAut(d8)] :-

35. : Source group d8 has generators: [(1,2,3,4), (1,3)]

36. : Range group = PermAut(d8) has generators: [(1,4)(2,6), (2,6)(3,5), (1,2,4,6)]

37. : Boundary homomorphism maps source generators to: [(1,4)(2,6), (2,6)(3,5)]

38. : Action homomorphism maps range generators to automorphisms:

39. (1,4)(2,6) --> { source gens --> [(1,2,3,4), (2,4)] }

40. (2,6)(3,5) --> { source gens --> [(1,4,3,2), (1,3)] }

41. (1,2,4,6) --> { source gens --> [(1,2,3,4), (1,4)(2,3)] }

42. These 3 automorphisms generate the group of automorphisms.

43. gap> RecFields(X2);

44. ["isDomain", "isParent", "source", "range", "boundary", "action", "aut", "isXMod", "operations", "name", "isAutomorphismXMod"]

45. Yukarıda vermiş olduğumuz örnekler de bilgisayara aktarılmış olup burada **CentralExtansionXMod**, **ConjugationXMod** ve **AutomorphismXMod** [10] örnekleri verilmiştir. Aşağıda, çaprazlanmış modül örneklerinde verilen Örnek 3.2.4. **CentralExtansionXMod** fonksiyonu yardımıyla hesaplanmıştır. Bu hesaplamada çaprazlanmış modülün boundary dönüşümü; *R*, *S* nin merkezinin alt grubunun çekirdeği olmak üzere *S* den *R* ye tanımlanan bir örten fonksiyon olarak alınmıştır [17].

gap> G:=Group((1,2)(3,4));

Group((1,2)(3,4))

gap> G.name:="G";

"G"

gap> t:=GroupHomomorphismByImages(G,G,G.generators,G.generators);

GroupHomomorphismByImages(G, G, [(1,2)(3,4)], [(1,2)(3,4)])

gap> IsHomomorphism(t);

true

gap> CX:=CentralExtensionXMod(t);

Crossed module [G->G]

gap> XModPrint(CX);

Crossed module [G->G] :-

: Source group G has generators: [(1,2)(3,4)]

: Range group = G has generators: [(1,2)(3,4)]

: Boundary homomorphism maps source generators to: [(1,2)(3,4)]

The automorphism group is trivial

gap> RecFields(CX);

["isDomain", "isParent", "source", "range", "boundary", "action", "aut", "isTrivialAction", "isXMod", "operations", "name", "isCentralExtensionXMod"]

46. Yine çaprazlanmış modül örneklerinden Örnek 3.2.3. te verilen çaprazlanmış modülün bilgisayar uygulaması **InnerAutomorphismXMod** fonksiyonu yardımıyla aşağıdaki biçimde yapılmıştır. Bunlardan elde edilen çaprazlanmış modül **AutomorphismXMod**(S,A) fonksi-yonundan elde edilen çaprazlanmış modüle denktir. Burada A, S nin iç otomorfizm grubudur.

gap>G:=Group((1,8,6,2)(3,5,7,4),(1,7,6,3)(2,5,8,4),(1,5,6,4)(2,3,8,7),(1,4,6,5)

(2,3,8,7),(1,6)(2,8)(3,7)(4,5));

Group((1,8,6,2)(3,5,7,4),(1,7,6,3)(2,5,8,4),(1,5,6,4)(2,3,8,7),(1,4,6,5)(2,3,8,7),

(1,6)(2,8)(3,7)(4,5))

gap> G.name:="G";

"G"

gap> X:=InnerAutomorphismXMod(G);

Crossed module [G->PermInn(G)]

gap> XModPrint(X);

Crossed module [G->PermInn(G)] :-

: Source group G has generators:

[(1,8,6,2)(3,5,7,4), (1,7,6,3)(2,5,8,4), (1,5,6,4)(2,3,8,7),(1,4,6,5)(2,3,8,7),(1,6)(2,8)(3,7)(4,5)]

: Range group = PermInn(G) has generators: [(3,6), (4,5), (2,8), (1,9)]

: Boundary homomorphism maps source generators to: [(3,6), (4,5), (2,8), (1,9), ()]

: Action homomorphism maps range generators to automorphisms:

(3,6)-->{sourcegens->[(1,8,6,2)(3,5,7,4),(1,7,6,3)(2,5,8,4),(1,5,6,4)(2,3,8,7), (1,5,6,4)(2,7,8,3),(1,6)(2,8)(3,7)(4,5)] }

(4,5) --> { source gens --> [(1,8,6,2)(3,5,7,4), (1,7,6,3)(2,5,8,4),

(1,4,6,5)(2,7,8,3), (1,4,6,5)(2,3,8,7), (1,6)(2,8)(3,7)(4,5)] }

(2,8) --> { source gens --> [(1,8,6,2)(3,5,7,4), (1,3,6,7)(2,4,8,5),

(1,5,6,4)(2,3,8,7), (1,4,6,5)(2,3,8,7), (1,6)(2,8)(3,7)(4,5)] }

(1,9) --> { source gens --> [(1,2,6,8)(3,4,7,5), (1,7,6,3)(2,5,8,4),

(1,5,6,4)(2,3,8,7), (1,4,6,5)(2,3,8,7), (1,6)(2,8)(3,7)(4,5)] }

These 4 automorphisms generate the group of automorphisms.

Ayrıca ileride kullanım formunu sunacağımız **XMod** fonksiyonu ile de bir çaprazlanmış modül üretmek mümkündür. Bu fonksiyonun kullanımına bir örnek aşağıdaki gibi verilebilir [17].

```
gap> s3c4 := Group((1,2),(2,3),(4,5,6,7));;
gap> s3c4.name := "s3c4";;
gap> s3 := Subgroup(s3c4, [(1,2), (2,3)]);;
gap> s3.name := "s3";;
gap> gen := s3c4.generators;;
gap> imb := [(1,2), (2,3), ()];;
gap> bX := GroupHomomorphismByImages(s3c4, s3, gen, imb);;
gap> # construct the inner automorphisms by (1,2) and (2,3)
gap> im1 := List(gen, g -> g^(1,2));;
gap> a1 := GroupHomomorphismByImages(s3c4, s3c4, gen, im1);;
gap> im2 := List(gen, g -> g^(2,3));;
gap> a2 := GroupHomomorphismByImages(s3c4, s3c4, gen, im2);;
gap> A := Group(a1, a2);;
gap> aX := GroupHomomorphismByImages(s3,A,[(1,2),(2,3)],[a1,a2]);;
gap> X := XMod(bX, aX);
```

Crossed module [s3c4->s3] |

2.2. Çaprazlanmış Modül Morfizmi

Tanım: 2.2.1 $X=(\partial:S\rightarrow R)$ ve $X^{'}=(\partial^{'}:S^{'}\rightarrow R^{'})$ iki çaprazlanmış modül olsun. Bu durumda, çaprazlanmış modül morfizmi

$$\langle\delta,p\rangle=X\rightarrow X^{'}$$

bir homomorfizm çiftidir ki burada $\delta:S\rightarrow S^{'}$ $p:R\rightarrow R^{'}$ ve aşağıdaki özellikleri sağlar.

$$\begin{tikzcd}
S \arrow[r, "\sigma"] \arrow[d, "\partial"'] & S' \arrow[d, "\partial'"] \\
R \arrow[r, "\rho"'] & R'
\end{tikzcd}$$

$1)\,\partial'\sigma(s)=p\partial(s)\quad ,\forall s\in S$

$2)\,\sigma\left(s^{r}\right)=\sigma(s)^{p(r)}$

X bir çaprazlanmış modül olmak üzere X' den X'e tüm morfizmlerin cümlesine X'in endomorfizmlerinin cümlesi denir ve $\mathrm{End}(X)$ ile gösterilir. Eğer δ *ve* ρ birlikte grup homomorfizmi ise bu durumda $\langle\delta,\rho\rangle$ çiftine bir izomorfizm denir.

47. X bir çaprazlanmış modül olmak üzere X' den X e tanımlı bütün morfizmlerin cümlesine X' in endomorfizmlerinin cümlesi denir ve **End(*X*)** ile gösterilir. Eğer δ ve ρ birlikte grup homomorfizmi ise; $\langle\delta,\rho\rangle$ çiftine bir izomorfizm denir.

GAP programı ile çaprazlanmış modül morfizmleri de hesaplanmıştır. Bu hesaplamalarda da yine yeni kayıt alanları oluşturulmuş olup bu kayıt alanları aşağıda verilmiştir. Bu kayıt alanları içerisinde yer alan **XModMorphismOps** işlemlerinin seti içerisinde fonksiyonun Monomorfizm, Epimiorfizm, İzomorfizm, Endomorfizm ve Otomorfizm olup olmadığını kontrol etmek için **IsMonomorphism**, **IsEpimorphism**, **IsIsomorphism**, **IsEndomorphism** ve **IsAutomorphism** fonksiyonları yer almaktadır. Bu kayıt alanları algoritmalarını vereceğimiz bilgisayar fonksiyonlarının algoritmalarında kullanılacaktır. Çaprazlanmış modüllerin morfizmlerini hesaplayabilmek için oluşturulan kayıt alanları aşağıda sunulmuştur [17].

Çaprazlanmış Modül Morfizmi Kayıt Alanları

mor.source,	X çaprazlanmış modülünün tanım grubu
mor.range,	Y çaprazlanmış modülünün değer grubu
mor.sourceHom,	X tanım grubundan Y tanım grubuna δ homomorfizmi
mor.rangeHom,	X değer grubundan Y değer grubuna ρ homomorfizmi
mor.isXModMorphism,	kontrol fonksiyonu, normalde doğru
mor.operations,	**XModMorphismOps** işlemlerinin özel bir seti
mor.name,	X ve Y isimlerinin birleştirilmesi

48. Bu fonksiyonun kullanılmasına bir örnek aşağıda verilmiştir.

49. gap> X;

50. Crossed module [s3->s3]

51. gap> t := GroupHomomorphismByImages(X.source,X.source,X.source.generators,X.source.generators);

52. GroupHomomorphismByImages(s3, s3, [(1,2), (2,3)], [(1,2), (2,3)])

53. gap> IsHomomorphism(t);

54. true

55. gap> h := GroupHomomorphismByImages(X.range,X.range,X.range.generators,X.range.generators);

56. GroupHomomorphismByImages(s3, s3, [(1,2), (2,3)], [(1,2), (2,3)])

57. gap> IsHomomorphism(h);

58. true

59. gap> mor:= XModMorphism(X,X,[t,h]);

60. Morphism of crossed modules <[s3->s3] >-> [s3->s3]>

61. gap> XModMorphismPrint(mor);

62. Morphism of crossed modules :-

63. : Source = Crossed module [s3->s3] with generating sets: [(1,2), (2,3)] [(1,2), (2,3)]

64. : Range = Source

65. : Source Homomorphism maps source generators to: [(1,2), (2,3)]

66. : Range Homomorphism maps range generators to: [(1,2), (2,3)]

67. gap> IsXModMorphism(mor)

68. gap> true

69. gap> RecFields(mor);

70. ["sourceHom", "rangeHom", "source", "range", "name", "domain", "isXModMorphism", "operations"]

71. gap> RecFields(mor.operations);

72. ["name", "operations", "IsMapping", "IsInjective", "IsSurjective", "IsBijection", "IsHomomorphism", "IsMonomorphism", "IsEpimorphism", "IsIsomorphism", "IsEndomorphism", "IsAutomorphism", "=", "<", "*", "/", "mod", "Comm", "^", "ImageElm", "ImagesElm", "ImagesSet", "ImagesSource", "ImagesRepresentative", "PreImageElm", "PreImagesElm", "PreImagesSet", "PreImagesRange", "PreImagesRepresentative", "CompositionMapping", "PowerMapping", "InverseMapping", "IsGroupHomomorphism",

73. "KernelGroupHomomorphism", "IsFieldHomomorphism", "KernelFieldHomomorphism",

74. "Order", "Print", "CompositeMorphism", "InverseMorphism", "Kernel"]

2.3. Geri Çekme (Pullback) Çaprazlanmış Modüller

75. Geri çekme (pullback) çaprazlanmış modüller ilk olarak Brown ve Higgins tarafından sunulmuştur.

Bir G- modülden H- modüle

$$f_*M = M \otimes_{ZG} ZH$$

şeklinde tanımlanan $f : G \to H$ morfizminin bir sol adjointe sahip olduğunu ve bu sol adjointin

$$f : (H - \text{modül}) \to (G - \text{modül})$$

bir pullback funktor olduğu gösterilmiştir [10].

76. Bu çalışmada; geri çekme çaprazlanmış modülün boundary dönüşümü ve etkisi iyi tanımlı olarak verilmiş olup bu tanımlama altında pullback çaprazlanmış modülü aşağıdaki gibi sunabiliriz.

77. X=($\partial : S \to R$) bir R–çaprazlanmış modül ve $\iota : Q \to R$ bir morfizm olsun. Bu durumda, X in ι yardımıyla oluşacak pullbacki

$$\iota^{**}X = (\partial^* : \iota^{**}S \to Q)$$

dir. Burada,

$$\iota^{**}S = \{(q,s) \in Q \times S \, : \, \iota q = \partial s\}$$

ve

$$\partial^*(q,s) = q$$

şeklinde tanımlanabilir. Q nun $\iota^{**}S$ üzerindeki sol etkisi ise

$${}^{q}(q_1,s) = (qq_1q^{-1}, {}^{\iota q}s)$$

şeklindedir.

$$\begin{tikzcd} \iota^{**}S \arrow[r] \arrow[d, "\partial^*"'] & S \arrow[d, "\partial"] \\ Q \arrow[r, "\iota"'] & R \end{tikzcd}$$

Bu etki ile tanımlanan,

$$\partial^* : \iota^{**}S \to Q$$

homomorfizmi bir çaprazlanmış modüldür [2].

Gerçekten de;

$\forall (q,s),(q',s') \in \iota^{**}S$ için

CM1) $\partial^*({}^{q'}(q,s)) = \partial^*({}^{q'}q, {}^{\iota q'}s)$

$= {}^{q'}q$

$= q'qq'^{-1}$

$= q'\partial^*(q,s)q'^{-1}$

CM2) $(q',s')(q,s)(q',s')^{-1} = (q',s')(q,s)(q'^{-1},s'^{-1})$

$= (q'qq'^{-1}, s'ss'^{-1})$

$= ({}^{q'}q, {}^{\partial s'}s)$

$= ({}^{q'}q, {}^{\iota q'}s)$

$= {}^{q'}(q,s)$

$= {}^{\partial^*(q',s')}(q,s)$

çaprazlanmış modül aksiyomları sağlanır. Dolayısıyla, ∂^* bir çaprazlanmış modüldür [2].

2.4. Cat^1-gruplar

Cat^1-gruplar 1982 yılında Loday tarafından tanımlandı. Loday ve Brown 1-Cat grup notasyonu için cat^1-gruplar notasyonunu kullandı.

Tanım 2.4.1 G bir grup *ve* $R \subset G$ bir alt grup olmak üzere $t,h : G \to R$ homorofizimleri

aşağıdaki aksiyomları sağlarsa bu durumda,

CAT1) $teh = h$ *ve* $het = t$

CAT2) $[\ker t, \ker h] = id_G = \{1_G\}$

$t^2 = t \quad tet = t$

$$h^2 = h \quad heh = h$$

$C = \left(e, t, h : G \to R\right)$ şeklinde gösterilen C ye bir cat[1]-grup denir. Burada *h* ve *t* yi başlangıç ve bitiş dönüşümleri olarak adlandıracağız.

$$G \underset{h}{\overset{t}{\rightrightarrows}} R$$

Tanım 2.4.2 $C = (e, t, h : G \to R)$ ve $C' = (e', t', h' : G' \to R')$ iki cat[1]-grup olsun. Bu durumda $C \to C'$ 'ne tanımlanan cat[1]-grup morfizmi, $\langle \gamma, p \rangle = C \to C'$ homomorfizmlerin-den oluşan $\langle \gamma, p \rangle$ çifti olup $\gamma : G \to G'$ ve $p : R \to R'$ şeklinde tanımlanan homomorfizmler aşağıdaki aksiyomları sağlar.

$h'\gamma = ph$

$t'\gamma = pt$

$$\begin{tikzcd}
G \arrow[r, "\gamma"] \arrow[d, shift right, "t"'] \arrow[d, shift left, "h"] & G' \arrow[d, shift right, "t'"'] \arrow[d, shift left, "h'"] \\
R \arrow[r, "\rho"'] \arrow[u, bend left=60, "e"] & R' \arrow[u, bend right=60, "e'"']
\end{tikzcd}$$

çaprazlanmış modüller kategorisi ile Cat[1]-grup kategorisi arasındaki eşdeğerlik Loday tarafından aşağıda verilen teorem ile ispatlandı.

Teorem 2.4.3 Aşağıda verilen önermeler birbirine denktir.

1. Çaprazlanmış modül $X = (\partial : S \to R)$

2. Cat[1]-grup $C = (e, t, h : G \to R)$

İspat: $(1 \Rightarrow 2)$ Kabul edelim ki $\partial : S \to R$ bir çaprazlanmış modül olsun. Bu durumda çaprazlanmış modül tanımı gereğince çaprazlanmış modül aksiyomları CM1) ve CM2) sağlanır.Bu şartlardan faydalanarak $Cat^1 - Grup$ aksiyomları olan CAT1) ve CAT2) aksiyomlarının sağlandığını göstermeliyiz.Bunun için $G = R \propto S$ grubu için başlangıç dönüşümü,

$$h(r,s) = r(\partial s)$$

Bitiş dönüşümü,

$$t(r,s) = r$$

ve embedding

$$e(r) = (r,1)$$

şeklinde tanımlansın. Bu durumda,

$$\begin{aligned} teh(r,s) &= te(r(\partial s)) \\ &= t(r(\partial s),1) \\ &= r(\partial s) \\ &= h(r,s) \end{aligned}$$

ve

$$\begin{aligned} het(r,s) &= he(r) \\ &= h(r,1) \\ &= r\partial 1 \\ &= r \\ &= t(r,s) \end{aligned}$$

olduğundan CAT1) aksiyomu sağlanır. Aynı şekilde CAT2) aksiyomunun sağlandığını

$$t(r,s) = r \qquad \ker t = (1,s)$$

$$\ker h = ((\partial s)^{-1}, s)$$

$$r\partial s = 1$$

$$r = (\partial s)^{-1}$$

$$\left[(1,s),(\partial s)^{-1},s\right] = (1,s)^{-1}\left[(\partial s)^{-1},s\right]^{-1}(1,s)((\partial s)^{-1},s)$$

$$= (1,s^{-1})\left[(\partial s),(s^{-1})^{\partial s}\right](1,s)((\partial s)^{-1},s)$$

$$= (\partial s, s^{-1^{\partial s}}(s^{-1})^{\partial s})((\partial s)^{-1}, s^{(\partial s)^{-1}}s)$$

$$= (\partial s, s^{-1}s^{-1}ss^{-1}s^{-1}s)((\partial s)^{-1}, sss^{-1}, s)$$

$$= (\partial s, s^{-2})((\partial s)^{-1}s^{2})$$

$$= ((\partial s)(\partial s)^{-1}, (s^{-2})^{(\partial s)^{-1}}s^{2})$$

$$= (1, ss^{-2}s^{-1}s^{2})$$

$$= (1, ss^{-2}s)$$

$$= (1,1)$$

şeklinde gösterebiliriz.

$(2 \Rightarrow 1)$ Kabul edelim ki $C = (e,t,h : G \to R)$ bir Cat^1-grup olsun. Göstereceğiz ki $\partial : S \to R$ bir çaprazlanmış modüldür. Bunun için CM1) ve CM2) şartlarının sağlandığının gösterilmesi yeterli olacaktır. O halde kabul edelim ki $x \in \ker t$ ve $y \in \ker h$ olsun. Bu durumda, $x \in (1,s)$ ve $y = (\partial r, r^{-1})$, $\forall r \in S$ ve böylece

$$[x,y] = xy = yx$$

olduğundan,

$$xy = (1,s)(\partial r, r^{-1})$$

$$= (\partial r, s^{\partial r}r^{-1})$$

$$= ((\partial r), r^{-1}srr^{-1})$$

$$= ((\partial r), r^{-1}s)$$

$$yx = (\partial r, r^{-1})(1, s)$$

$$= (\partial r, r^{-1}s)$$

$$s^{\partial r} r^{-1} = r^{-1}s \Rightarrow s^{\partial r} = r^{-1}sr$$

olur. Dolayısıyla CM2) aksiyomu sağlanır.

Cat[1]-grupların bilgisayar hesaplamalarında, cat[1]-grup $C = (e; t, h : G \to R)$ için GAP programında var olan kayıt alanları oluşturulmuş ve bu kayıt alanları GAP ortak paketi olan XMOD [7] da aşağıdaki gibi sunulmuştur [17].

Bu kayıt alanlarında verilen **Cat1GroupOps** işlemlerinin özel setleri ve cat[1]-gruplarda yapılan işlemlerde **C.operations** kayıt alanında verilmektedir.

Cat[1]-gruplar; GAP programı aracılığıyla iki şekilde elde edilir ki bunlar;

- **Cat1** fonksiyonunu kullanılarak
- Denk olduğu kategori kullanılarak

Cat^1-Gruplar için Kayıt Alanları

C.source,	*G*, tanım grubu
C.range,	*R*, değer grubu
C.tail,	*t,* bitiş homomorfizmi
C.head,	*h,* başlangıç homomorfizmi
C.embedRange,	*e, R* nin *G* deki gömme homomorfizmi
C.isCat1Group,	kontrol fonksiyonu, normalde doğru
C.isDomain,	daima doğru
C.kernel,	*t* nin çekirdeğine izomorfik olan permütasyon grubu
C.boundary, ∂,	*t* nin çekirdeğe kısıtlaması
C.embedKernel, $\iota : S \to Ker\ t$	izomorfizm
C.operations,	**Cat1GroupOps** işlemlerinin özel bir seti
C.name,	*G* ve *R* isimlerinin birleştirilmesi

Cat1 fonksiyonunun kullanılması bilinen grup ve homomorfizmlerden cat^1-grup oluşturmak için geliştirilmiştir. Denk kategori olan çaprazlanmış modüllerden bir cat^1-grup oluşturmak için de **Cat1XMod** fonksiyonu geliştirilmiştir. Bu fonksiyonların kullanılmasına ilişkin örnekler aşağıda verilmiştir [17].

```
gap> G:=Group((1,2),(3,4));
Group( (1,2), (3,4) )
gap> G.name:="k4";
"k4"
gap> gen:= G.generators;[ (1,2), (3,4) ]
gap> t:=GroupHomomorphismByImages(G,G,gen,gen);
GroupHomomorphismByImages( k4, k4, [ (1,2), (3,4) ], [ (1,2), (3,4) ] )
gap> IsHomomorphism(t);
true
gap> C:=Cat1(G, t, t);
cat1-group [k4 ==> k4]
gap> Cat1Print(C);
cat1-group [k4 ==> k4] :-
: source group has generators:  [ (1,2), (3,4) ]
:  range group has generators:  [ (1,2), (3,4) ]
: tail homomorphism maps source generators to:  [ (1,2), (3,4) ]
: head homomorphism maps source generators to:  [ (1,2), (3,4) ]
: range embedding maps range generators to:  [ (1,2), (3,4) ]
: kernel has generators:  [  ]
: boundary homomorphism maps generators of kernel to: [  ]
: kernel embedding maps generators of kernel to:  [  ]
gap> RecFields(C);
```

["source", "range", "tail", "head", "embedRange", "kernel", "boundary", "embedKernel", "isCat1", "isDomain", "operations", "name", "isIdentity"]

gap> RecFields(C.operations);

["name", "operations", "Elements", "IsFinite", "Size", "=", "<", "in",

"IsSubset", "Intersection", "Union", "IsParent", "Parent", "Difference", "Representative", "Random", "Print", "InclusionMorphism", "WhiteheadPermGroup", "Actor", "InnerActor"]

Cat1 gruplar kategorisi ile çaprazlanmış modüller kategorisi arasındaki denklik GAP programı aracılığıyla hesaplanmış olup hesaplama örneği aşağıda sunulmuştur.

gap> X:=XModSelect(24,11,"conj");

Crossed module [d8|Xc3->d8|Xc3]

gap> C:=Cat1XMod(X);

cat1-group [Perm(d8|Xc3 |X d8|Xc3) ==> d8|Xc3]

gap> Cat1Print(C);

cat1-group [Perm(d8|Xc3 |X d8|Xc3) ==> d8|Xc3] :-

: source group has generators: [(7, 9, 8)(10,12,11)(19,21,20)(22,24,23)(29,30,31),

(2, 3)(4,13)(5,15)(6,14)(7,22)(8,24)(9,23)(11,12)(17,18)(20,21) (25,26,27,28)(30,31)

,(7,22)(8,23)(9,24)(10,19)(11,20)(12,21)(26,28),(1,2,3)(4,5,6)(7,8,9)(10,11,12)(13,14,1
5)

(16,17,18)(19,20,21)(22,23,24), (1,10,16,19)(2,12,17,21)(3,11,18,20)(4, 7,13,22)

(5, 9,14,24)(6, 8,15,23), (1, 4)(2, 5)(3, 6)(7,19)(8,20)(9,21)

(10,22)(11,23)(12,24)(13,16)(14,17)(15,18)]

: range group has generators: [(5,6,7), (1,2,3,4)(6,7), (2,4)]

: tail homomorphism maps source generators to: [(5, 6, 7), (1, 2, 3, 4)(6, 7), (2, 4), (), (), ()]

: head homomorphism maps source generators to:[(5, 6, 7), (1, 2, 3, 4)(6, 7), (2, 4), (5, 6, 7),

(1, 2, 3, 4)(6, 7), (2, 4)]

: range embedding maps range generators to:

[(7, 9, 8)(10,12,11)(19,21,20)(22,24,23)(29,30,31), (2, 3)(4,13)(5,15)(6,14)(7,22)(8,24)

(9,23)(11,12)(17,18)(20,21)(25,26,27,28)(30,31),(7,22)(8,23)(9,24)(10,19)(11,20)

(12,21)(26,28)]

: kernel has generators: [(5,6,7), (1,2,3,4)(6,7), (2,4)]

: boundary homomorphism maps generators of kernel to: [(5,6,7), (1,2,3,4)(6,7), (2,4)]

: kernel embedding maps generators of kernel to:

[(1, 2, 3)(4, 5, 6)(7, 8, 9)(10,11,12)(13,14,15)(16,17,18)(19,20,21)(22,23,24), (1,10,16,19)

(2,12,17,21)(3,11,18,20)(4, 7,13,22) (5, 9,14,24)(6, 8,15,23), (1, 4)(2, 5)(3, 6)(7,19)

(8,20)(9,21) (10,22)(11,23)(12,24)(13,16)(14,17)(15,18)]

: associated crossed module is Crossed module [d8|Xc3->d8|Xc3]

2.5. Cat^1 -Grup Morfizmi

$C=(e;t,h:S\rightarrow R)$ ve $C'=(e';t',h':S'\rightarrow R')$ iki cat^1-grup olsun. Bu durumda $C\rightarrow C'$ ne tanımlanan cat^1-grup morfizmi $\langle\delta,\rho\rangle:C\rightarrow C'$ homorfizmlerinden oluşan $\langle\delta,\rho\rangle$ çifti olup $\delta:S\rightarrow S'$ ve $\rho:R\rightarrow R'$ şeklinde tanımlanan homomorfizmler aşağıdaki aksiyomları sağlar.

$$\begin{tikzcd} S \arrow[r, "\delta"] \arrow[d, shift right, "t"'] \arrow[d, shift left, "h"] & S' \arrow[d, shift right, "t'"'] \arrow[d, shift left, "h'"] \\ R \arrow[r, "\rho"'] \arrow[u, bend left=60, "e"] & R' \arrow[u, bend right=60, "e'"'] \end{tikzcd}$$

$h'\delta = \rho h$

$t'\delta = \rho t$

$e'\rho = \delta e$

$s \in S,\ s' \in S',\ r \in R,\ r' \in R'$ olmak üzere

$\delta(s) = s',\quad h'(s') = r'$ ve $s \in S,\ \ r \in R,\ r' \in R'$

$h(s) = r,\quad \rho(r) = r' \quad \Rightarrow \quad h'\delta(s) = h's' = r' = \rho(r) = \rho h(s)$

$\Rightarrow \quad t'\delta(s) = t's' = r' = \rho(r) = \rho t(s)$

$s \in S,\ \ r \in R,\ r' \in R' \quad \Rightarrow \quad e'\rho(r) = e'(r') = s' = \delta(s) = \delta e(r)$

olup morfizm şartları sağlanır.

GAP programı aracılığıyla cat^1-grup morfizmleride hesaplanmıştır. Bu hesaplamada da çaprazlanmış modül morfizminde olduğu gibi cat^1-grup morfizmi kayıt alanları oluşturulmuştur. Bu kayıt alanları içerisinde yer alan **Cat1MorphismOps** işlemlerinin bir seti içerisinde cat^1-grup morfizminin monomorfizm, epimorfizm, izomorfizm, endomorfizm ve otomorfizm olup olmadığını kontrol etmek için bilgisayar fonksiyonları geliştirilmiştir [17].

Cat^1-grup morfizmi $mor = (\delta, \rho) : C \to D$ için geliştirilen kayıt alanı

Cat^1-Grup Morfizmi için Kayıt Alanları

mor.source, Cat^1-grubun tanım grubu, C

mor.range, Cat^1-grubun değer grubu, D

mor.sourceHom, C nin tanım grubundan D δ nin tanım grubuna homomorfizmi

mor.rangeHom, C nin değer grubundan D ρ nin değer grubuna homomorfizmi

mor.isCat1Morphism, kontrol fonksiyonu, normalde doğru

mor.operations, **Cat1MorphismOps** işlemlerinin özel bir seti

mor.name, C ve D isimlerinin birleştirilmesi biçimindedir. Bu kayıt alanlarını ve bilgisayar fonksiyonunun kullanılmasına ilişkin aşağıdaki örneği verelim.

```
gap> C;
cat1-group [c12 ==> c3]
gap>t:=GroupHomomorphismByImages
(C.source,C.source,C.source.generators,C.source.generators);
GroupHomomorphismByImages( c12, c12, [ (1,2,3,4), (5,6,7) ], [ (1,2,3,4), (5,6,7) ] )
gap> IsHomomorphism(t);
true
gap>h:=GroupHomomorphismByImages
(C.range,C.range,C.range.generators,C.range.generators);
GroupHomomorphismByImages( c3, c3, [ (5,6,7) ], [ (5,6,7) ] )
gap> IsHomomorphism(h);
true
gap> mor:=Cat1Morphism(C,C,[t,h]);
Morphism of cat1-groups <[c12 ==> c3]-->[c12 ==> c3]>
gap> Cat1MorphismPrint(mor);
Morphism of cat1-groups :=
: Source = cat1-group [c12 ==> c3]
:  Range = cat1-group [c12 ==> c3]
: Source homomorphism maps source generators to:  [ (1,2,3,4), (5,6,7) ]
: Range homomorphism maps range generators to:  [ (5,6,7) ]
gap> RecFields(mor);
[ "source", "range", "sourceHom", "rangeHom", "domain",
"isCat1Morphism",  "operations", "name" ]
```

gap> RecFields(mor.operations);

["name", "operations", "IsMapping", "IsInjective", "IsSurjective", "IsBijection", "IsHomomorphism", "IsMonomorphism", "IsEpimorphism", "IsIsomorphism", "IsEndomorphism", "IsAutomorphism", "=", "<", "*", "/", "mod", "Comm", "^", "ImageElm", "ImagesElm", "ImagesSet", "ImagesSource", "ImagesRepresentative", "PreImageElm", "PreImagesElm", "PreImagesSet", "PreImagesRange", "PreImagesRepresentative", "CompositionMapping", "PowerMapping", "InverseMapping", "IsGroupHomomorphism",

"KernelGroupHomomorphism", "Print", "CompositeMorphism", "InverseMorphism"]

2.6. Geri Çekme (Pullback) $\mathbf{Cat^1}$-Gruplar

Çaprazlanmış modüller kategorisi ile cat^1 gruplar kategorisi arasındaki denklik kullanılarak, pullback cat^1 gruplar Alp tarafından, aşağıdaki şekilde tanımlanmıştır [2].

$C=(e;t,h:G\to R)$ ve $\iota:Q\to R$ bir grup homomorfizmi olmak üzere G nin ι morfizmi aracılığıyla oluşturulan

$$\iota^{**}C=(e^{**};t^{**},h^{**}:\iota^{**}G\to Q)$$

geriçekmesi

$$\iota^{**}G=\{(q_1,g,q_2)\in Q\times G\times Q\ :\ \iota q_1=tg,\ \iota q_2=hg\}$$

olmak üzere,

$t^{**}(q_1,g,q_2)=q_1$

$h^{**}(q_1,g,q_2)=q_2$

ve

$e^{**}(q)=(q,e\iota q,q)$

şeklinde tanımlayalım.

$$\begin{tikzcd}
\iota^{**}G \arrow[rr, "h^{**}"] \arrow[dr, "\pi"] \arrow[dd, "t^{**}"'] & & Q \arrow[dr, "\iota"] \arrow[ll, bend right, "e^{**}"'] & \\
& G \arrow[rr, "h"'] \arrow[dd, "t"] & & R \arrow[ll, bend right, "e"'] \\
Q \arrow[dr, "\iota"'] \arrow[uu, bend left, "e^{**}"] & & & \\
& R \arrow[uu, bend left, "e"] & &
\end{tikzcd}$$

Aynı zamanda (π, ι) bir Cat^1- grup morfizmi olup

$$\pi \ : \ \iota^{**}G \to G$$

$$(q_1, g, q_2) \mapsto \text{g}$$

şeklinde tanımlanır. Yukarıda tanımlanan $\iota^{**}G$ cümlesi t^{**}, h^{**}, e^{**} tanımı ile birlikte bir Cat^1-grup oluşturur. Gerçekten de,

CAT1) $het(q_1, g, q_2) = he(q_1)$

$$= h(q_1, e\iota q_1, q_1)$$

$$= q_1$$

$$teh(q_1, g, q_2) = te(q_2)$$

$$= t(q_2, e\iota q_2, q_2)$$

$$= q_2$$

CAT2) $x = (q_1^{'}, g_1, q_2) \in Ker\ t^{**}$

$$y = (q_1, g_2, q_2^{'}) \in Ker\ h^{**}$$

olduğunu kabul edelim. Bu durumda ι^{**} tanımından dolayı $q_1^{'} = q_2^{'} = 1$ dir. Böylece, $g_1 \in Ker\ t^{**}, g_2 \in Ker\ h^{**}$ olur ve

$$[x,y]=(1_Q,[q_1,q_2],1_Q)$$

$$=(1_Q,1_G,1_Q)$$

olduğundan, $[Ker\ t^{**}, Ker\ h^{**}]=1$ dir.

3. LİE CEBİRLERİ ÜZERİNDE ÇAPRAZLANMIŞ MODÜLLER

Bu bölümde, birinci bölümde vermiş olduğumuz kategorilerin Lie cebirleri üzerindeki uygulamaları verilecektir. Lie cebirleri verildikten sonra Lie cebirleri üzerindeki çaprazlanmış modül kavramı, çaprazlanmış modül morfizmi ve örnekleri de bu bölüm içerisinde verilmiştir. Ayrıca lie cebirleri için çaprazlanmış alt modüller bu bölüm içerisinde örnekleriyle birlikte detaylı olarak verilmiştir. Sonrasında Loday`in denkliğini göstermiş olduğu Cat^1-Lie cebir kategorisi de bu bölümde verilmiştir.

3.1 Lie Cebirleri

Tanım 3.1.1 A birimli ve değişmeli halka ve M bir A-modül olsun. Eğer $\forall\ x, y, z \in \mathrm{M}$ için,

i-) $[x,x]=0$

ii-) $[x,[y,z]]+[y,[z,x]]+[z,[x,y]]=0$ (Jakobi özelliği)

özelliklerini sağlayan bir $[,]:\mathrm{M}\times\mathrm{M}\to\mathrm{M}$ bilineer fonksiyonu varsa M ye A üzerinde bir Lie Cebiri denir. $[,]$ fonksiyonuna da Lie braketi ya da Lie çarpımı denir. A üzerindeki Lie cebiri bir A-bilineer dönüşümü ve M cismi ile birlikte A-modüldür. Yukarıda verilen (i),(ii) ve bilineerlik özellikleri gereğince

$$0=[x+y,x+y]=[x,y]+[y,x]$$

olduğundan,

iii-) $[x,y]=-[y,x]$

iv-) $[x,[y,z]]=[[x,y],z]+[y,[x,z]]$

eşitlikleri gösterilir [9].

Örnek 3.1.2 M, A üzerinde bir cebir olsun. $[x,y]=xy-yx$ şeklinde tanımlanan $[,]:\mathrm{M}\times\mathrm{M}\to\mathrm{M}$ fonksiyonunu alalım.

i-) $[x,x]=xx-xx=0$

ii-)

$$\left[x,[y,z]\right]+\left[y,[z,x]\right]+\left[z,[x,y]\right]=[x,yz-zy]+[y,zx-xy]+[z,xy-yx]$$

=x(yz-zy)-(yz-zy)x+y(zx-xz)-(zx-xz)y +z(xy-yx)-(xy-yx)z $=0$

dır. $[,]$ fonksiyonunun bilineer olduğu da kolayca gösterilir. Böylece M , $[,]$ ile birlikte bir Lie cebiridir [9].

Örnek 3.1.3 M bir Lie cebiri olsun. $\forall \; x,y\in \mathrm{M}$ için $[x,y]=0$ oluyorsa M abelyen Lie cebiridir [9].

i-) $[x,x]=0$

ii-) $\left[x,[y,z]\right]+\left[y,[z,x]\right]+\left[z,[x,y]\right]=0$

Tanım 3.1.4 $\mathrm{M}_1,\mathrm{M}_2$ A üzerinde iki Lie cebiri ve $\partial:\mathrm{M}_1\to\mathrm{M}_2$ bir A-modül homomorfizmi olsun. $\forall \; x,y \in \mathrm{M}_1$ için $\partial([x,y])=[\partial(x),\partial(y)]$ ise ∂ ya bir Lie cebir homomorfizmi denir.

Örnek 3.1.5 M,A üzerinde bir cebir olsun. $[x,y]=xy-yx$ şeklinde tanımlanan $[,]:\mathrm{M}\times\mathrm{M}\to\mathrm{M}$ fonksiyonu bir Lie cebiridir.

Lie cebirlerinin GAP [34] programı aracılığıyla hesaplanabilir olduğu GAP programı ortak paketleri LAGUNA ve LieAlgDB tarafından yapılabilmektedir. LAGUNA paketinde [21] Lie cebirlerini hesaplayabilmek için geliştirilen fonksiyonlar aşağıda detaylı ve LAGUNA paketinden alınmış örnekleriyle verilmektedir.

LieAlgebraByDomain(A)

Bu fonksiyon girdi olarak grup cebiri alarak çıktı olarak Lie cebir vermektedir. Burada çarpım işlemi olarak Lie braketi [a, b] = ab-ba alınmıştır. Bu fonksiyonun kullanımı LAGUNA paketini oluşturanlar tarafından önerilmemektedir. Bunun nedeni, asosyatif cebirlerden rahatlıkla LieAlgebra(A) fonksiyonunun kullanılmasıyla Lie cebirlerinin elde edilmesidir. LieAlgebra fonksiyonu da LieAlgebraByDomain fonksiyonunu kullandığından bu fonksiyonun iki kere kullanılmasından kaçınmak için bu fonksiyonun kullanılması önerilmemektedir [13].

```
gap> G := SymmetricGroup(3);; FG := GroupRing( GF( 2 ), G );
```

<algebra-with-one over GF(2), with 2 generators>

gap> L := LieAlgebra(FG);

#I LAGUNA package: Constructing Lie algebra ...

<Lie algebra over GF(2)>

IsLieAlgebraByAssociativeAlgebra (L)

Bu fonsiyon asosyatif cebirden elde edilen bir L Lie cebirini verir ve Lie Cebir olup olmadığını test eder [13].

gap> KG := GroupRing(GF(3), DihedralGroup(16));

<algebra-with-one over GF(3), with 4 generators>

gap> L := LieAlgebra (KG);

#I LAGUNA package: Constructing Lie algebra ...

<Lie algebra over GF(3)>

gap> IsLieAlgebraByAssociativeAlgebra(L);

true

UnderlyingAssociativeAlgebra (L)

Lie cebirlerin underlying asosyatif cebirlerine döner ve eğer bir asosyatif cebirden bir L Lie cebiri elde edilmişse bunun mutlaka bir underlying asosyatif cebir olduğunu algılar [13].

gap> KG := GroupRing(GF(2), DihedralGroup(16));

<algebra-with-one over GF(2), with 4 generators>

gap> L := LieAlgebra (KG);

#I LAGUNA package: Constructing Lie algebra ...

<Lie algebra over GF(2)>

gap> UnderlyingAssociativeAlgebra(L);

<algebra-with-one over GF(2), with 4 generators>

gap> last = KG;

```
true
```

NaturalBijectionToLieAlgebra(FG)

Bu fonksiyon bir dönüşüme döner ve bir asosyatif cebir A nın underlying vektör uzayı ile bundan üretilmiş Lie cebiri arasındaki doğal lineer dönüşümdür. Bu iki cebir arasındaki vektör uzay izomorfizmidir bir cebir izomorfizmi değildir [13].

```
gap> F := GF( 2 ); G := SymmetricGroup( 3 ); FG := GroupRing( F, G );
GF(2)
Sym( [ 1 .. 3 ] )
<algebra-with-one over GF(2), with 2 generators>
gap> t := NaturalBijectionToLieAlgebra( FG );
MappingByFunction( <algebra-with-one over GF(2), with
2 generators>, <Lie algebra over GF(
2)>, <Operation "LieObject">, function( y ) ... end )
gap> a := Random( FG );
(Z(2)^0)*(1,2,3)+(Z(2)^0)*(1,3,2)+(Z(2)^0)*(1,3)
gap> a * a; # product in the associative algebra
(Z(2)^0)*()+(Z(2)^0)*(1,2,3)+(Z(2)^0)*(1,3,2)
gap> b := a^t;
LieObject( (Z(2)^0)*(1,2,3)+(Z(2)^0)*(1,3,2)+(Z(2)^0)*(1,3) )
gap> b * b; # product in the Lie algebra (commutator) ...
LieObject( <zero> of ... ) # ... must be zero!
```

NaturalBijectionToAssociativeAlgebra (L)

Biraz önceki lineer dönüşümün tersidir [13].

```
gap> G := SymmetricGroup(3); FG := GroupRing( GF( 2 ), G );
Sym( [ 1 .. 3 ] )
```

<algebra-with-one over GF(2), with 2 generators>

gap> L := LieAlgebra(FG);

#I LAGUNA package: Constructing Lie algebra ...

<Lie algebra over GF(2)>

gap> s := NaturalBijectionToAssociativeAlgebra(L);

MappingByFunction(<Lie algebra over GF(2)>, <algebra-with-one over GF(

2), with 2 generators>, function(y) ... end, <Operation "LieObject">)

gap> InverseGeneralMapping(s) = NaturalBijectionToLieAlgebra(FG);true

IsLieAlgebraOfGroupRing (L)

Eğer bir asosyatif cebirden bir L Lie cebiri elde edilmişse bunun bir grup halkası olması kaçınılmazdır. Bu nedenle Lie cebir fonksiyonunun bir özelliği olarak kayıt altında tutulması gerekli olan bir özelliktir [13].

gap> F := GF(2); G := SymmetricGroup(3); FG := GroupRing(F, G);

GF(2)

Sym([1 .. 3])

<algebra-with-one over GF(2), with 2 generators>

gap> L := LieAlgebra(FG);

#I LAGUNA package: Constructing Lie algebra ...

<Lie algebra over GF(2)>

gap> IsLieAlgebraOfGroupRing(L);

true

UnderlyingGroup (L)

Underlying grup`a döner. Bir L Lie cebirinin underlying grubu bir grup halkasından elde edilir ve bu grup halkasının underlying grubu olarak tanımlanır [13].

gap> F := GF(2); G := SymmetricGroup(3); FG := GroupRing(F, G);

GF(2)

```
Sym( [ 1 .. 3 ] )

<algebra-with-one over GF(2), with 2 generators>

gap> L := LieAlgebra( FG );

#I LAGUNA package: Constructing Lie algebra ...

<Lie algebra over GF(2)>

gap> UnderlyingGroup( L );

Sym( [ 1 .. 3 ] )

gap> LeftActingDomain( L );

GF(2)
```

Embedding (G, L)

İki dönüşümün birleşiminden oluşan bir dönüşüm oluşturur. Bunun için FG bir grup halkası, U, G`nin bir altmagması ve L, FG`den üretilen Lie cebir olsun. Böylece Embedding(U,L), U dan L`ye var olan dönüşümü ütretir [13].

```
gap> F := GF( 2 ); G := SymmetricGroup( 3 ); FG := GroupRing( F, G );

GF(2)

Sym( [ 1 .. 3 ] )

<algebra-with-one over GF(2), with 2 generators>

gap> L := LieAlgebra( FG );

#I LAGUNA package: Constructing Lie algebra ...

<Lie algebra over GF(2)>

gap> f := Embedding( G, L );

CompositionMapping( MappingByFunction( <algebra-with-one over GF(2), with

2 generators>, <Lie algebra over GF(

2)>, <Operation "LieObject">, function( y ) ... end ), <mapping: SymmetricGrou\

p( [ 1 .. 3 ] ) -> AlgebraWithOne( GF(2), ... ) > )
```

gap> (1,2)^f + (1,3)^f;

LieObject((Z(2)^0)*(1,2)+(Z(2)^0)*(1,3))

LieCentre (L)

Bu fonksiyon bir Lie cebir üretir. Grup halkası tarafından oluşturulan Lie cebirin merkezi underlying grup halkasının merkezine karşılık gelir ve bu hesaplama grubun denklik sınıfları aracılığıyla yapılır. Bu metod bu fikrin kullanılmasıyla L`nin merkezine döner [13].

gap> G := SmallGroup(256, 400); FG := GroupRing(GF(2), G);

<pc group of size 256 with 8 generators>

<algebra-with-one over GF(2), with 8 generators>

gap> L := LieAlgebra(FG);

#I LAGUNA package: Constructing Lie algebra ...

<Lie algebra over GF(2)>

gap> C := LieCentre(L);

<Lie algebra of dimension 28 over GF(2)>

gap> D := LieDerivedSubalgebra(L);

#I LAGUNA package: Computing the Lie derived subalgebra ...

<Lie algebra of dimension 228 over GF(2)>

gap> c := Dimension(C); d := Dimension(D); l := Dimension(L);

28

228

256

gap> c + d = l;

true # This is always the case for Lie algebras of group algebras!

LieDerivedSubalgebra (L)

Bu fonksiyon bir Lie cebirine döner. Eğer L Lie cebiri bir grup halkasından oluşmuşsa bu durumda bu metod türetilen L`nin alt cebirlerine döner [13].

```
gap> G := SmallGroup( 256, 400 ); FG := GroupRing( GF( 2 ), G );
<pc group of size 256 with 8 generators>
<algebra-with-one over GF(2), with 8 generators>
gap> L := LieAlgebra( FG );
#I LAGUNA package: Constructing Lie algebra ...
<Lie algebra over GF(2)>
gap> C := LieCentre( L );
<Lie algebra of dimension 28 over GF(2)>
gap> D := LieDerivedSubalgebra( L );
#I LAGUNA package: Computing the Lie derived subalgebra ...
<Lie algebra of dimension 228 over GF(2)>
gap> l := Dimension( L ); c := Dimension( C ); d := Dimension( D );
256
28
228
gap> c + d = l;
true # This is always the case for Lie algebras of group algebras!
```

IsLieAbelian (L)

Bir asosyatif cebir A`dan üretilen L Lie cebirinin abelyen olması için gerek ve yeter koşul A`nın abelyen olmasıdır. Dolayısıyla, bu metod IsAbelian(A) yı oluşturur [13].

```
gap> G := SymmetricGroup( 3 ); FG := GroupRing( GF( 2 ), G);
Sym( [ 1 .. 3 ] )
<algebra-with-one over GF(2), with 2 generators>
gap> L := LieAlgebra( FG );
#I LAGUNA package: Constructing Lie algebra ...
```

<Lie algebra over GF(2)>

gap> IsAbelian(G);

false

gap> IsAbelian(L); # This command should never be used for Lie algebras!

true # It gives a result, but (probably) not the desired one.

gap> IsLieAbelian(L); # Instead, IsLieAbelian is the correct command.

False

IsLieSolvable (L)

Lie komutatörlerini hesaplamak için kullanılan Passi, Passman ve Sehgal`ın geliştirmiş oldukları grupların sınıflandırılması yönetimini kullanmaktadır. Bu sınıflandırmada grup halkaların solvable olması durumunda elde edilen Lie cebirlerine döner [13].

gap> G := SmallGroup(256, 400); FG := GroupRing(GF(2), G);

<pc group of size 256 with 8 generators>

<algebra-with-one over GF(2), with 8 generators>

gap> L := LieAlgebra(FG);

#I LAGUNA package: Constructing Lie algebra ...

<Lie algebra over GF(2)>

gap> IsLieSolvable(L); # This is very fast.

true

gap> List(LieDerivedSeries(L), Dimension); # This is very slow.

[256, 228, 189, 71, 0]

IsLieNilpotent (L)

Lie komutatörlerini hesaplamak için kullanılan Passi, Passman ve Sehgal`ın geliştirmiş oldukları grupların sınıflandırılması yönetimini kullanmaktadır. Bu sınıflandırmada grup halkaların Lie nilpotent olması durumunda elde edilen Lie cebirlerine döner [13].

gap> G := SmallGroup(256, 400); FG := GroupRing(GF(2), G);

```
<pc group of size 256 with 8 generators>
<algebra-with-one over GF(2), with 8 generators>
gap> L := LieAlgebra( FG );
#I LAGUNA package: Constructing Lie algebra ...
<Lie algebra over GF(2)>
gap> IsLieNilpotent( L ); # This is very fast.
true
gap> List( LieLowerCentralSeries( L ), Dimension ); # This is very slow.
[ 256, 228, 222, 210, 191, 167, 138, 107, 76, 54, 29, 15, 6, 0 ]
```

IsLieMetabelian (L)

Lie komutatörlerini hesaplamak için kullanılan Levin ve Rosenberger`in geliştirmiş oldukları grupların sınıflandırılması yönetimini kullanmaktadır. Bu sınıflandırmada grup halkaların Lie metabelian olması durumunda elde edilen Lie cebirlerine döner [13].

```
gap> G := SmallGroup( 256, 400 ); FG := GroupRing( GF( 2 ), G );
<pc group of size 256 with 8 generators>
<algebra-with-one over GF(2), with 8 generators>
gap> L := LieAlgebra( FG );
#I LAGUNA package: Constructing Lie algebra ...
<Lie algebra over GF(2)>
gap> IsLieMetabelian( L );
false
```

IsLieCentreByMetabelian (L)

Lie komutatörlerini hesaplamak için kullanılan Rosmanith`in geliştirmiş olduğu grupların sınıflandırılması yönetimini kullanmaktadır. Bu sınıflandırmada grup halkaların Lie centre by metabelian olması durumunda elde edilen Lie cebirlerine döner [13].

```
gap> G := SymmetricGroup( 3 ); FG := GroupRing( GF( 2 ), G );
```

Sym([1 .. 3])

<algebra-with-one over GF(2), with 2 generators>

gap> L := LieAlgebra(FG);

#I LAGUNA package: Constructing Lie algebra ...

<Lie algebra over GF(2)>

gap> IsLieMetabelian(L);

false

gap> IsLieCentreByMetabelian(L);

true

3.2. Çaprazlanmış Modüller

Tanım 3.2.1 M ve P Lie cebirleri olsun. $\mathrm{P}\times\mathrm{M}\to\mathrm{M}$, (p,m) $\to {}^{p}m=[p,m]$ dönüşümünün P nin M üzerindeki Lie etkisi olması için gerek ve yeter şart $\forall$ a $\in A$,

$m, m_1 \in \mathrm{M}$,p, $p_1 \in \mathrm{P}$ *için* ,

Lie Act 1: ${}^{ap}m = {}^{p}(am) = a^{pm}$

Lie Act 2: ${}^{p}(m+m_1) = {}^{p}m + {}^{p}m_1$

Lie Act 3: ${}^{(p+p_1)}m = {}^{p}m + {}^{p_1}m$

Lie Act 4: $m^{\,(p,p_1)} = {}^{p}({}^{p_1}m) - {}^{p_1}({}^{p}m)$

Lie Act 5: ${}^{p}[m,m_1] = [m^{p},m_1] + [m,{}^{p}m_1]$

eşitliklerinin sağlanmasıdır [16].

Tanım 3.2.2 M ve P iki Lie cebir, $\partial:\mathrm{M}\to\mathrm{P}$ Lie cebir homomorfizmi ve P nin M üzerine Lie etkisi ile birlikte $\forall m, m_1 \in \mathrm{M}$ ve $p\in\mathrm{P}$ için,

Lie CM1: $\partial^{p}(m) = [p,\partial_m]$

Lie CM2: $^{(\partial m)}m_1 = \left[m, m_1 \right]$ (Peiffer özdeşliği)

şartları sağlanıyorsa $(\mathrm{M}, \mathrm{P}, \partial)$ üçlüsüne Lie çaprazlanmış P-modül denir. Sadece Lie CM1 şartını sağlayan $(\mathrm{M}, \mathrm{P}, \partial)$ üçlüsüne Lie yarı çaprazlanmış P-modül denir [16].

3.3. Lie Cebirlerin Çaprazlanmış Modüllerinin Standart Örnekleri

Örnek 3.3.1 M bir Lie cebiri, I da M nin bir ideali olsun.

$$\partial : \mathrm{I} \to \mathrm{M}$$

$$i \to i$$

içine dönüşümünü ele alalım. M nin I üzerine etkisi

$$\mathrm{M} \times \mathrm{I} \to \mathrm{I}$$

$$(m, i) \to \left[m, i \right]$$

Lie çarpımı şeklinde verilsin.

Lie CM1: $\partial(m, i) = \partial\left[m, i \right]$

$= \left[m, i \right]$

$= \left[m, \partial i \right]$

LieCM2: $\partial m_{m^{'}} = m_{m^{'}}$

$= \left[m, m^{'} \right]$

şartları sağlandığından dolayı $(\mathrm{I}, \mathrm{M}, \partial)$ üçlüsü çaprazlanmış modül yapısı oluşturur [9].

Örnek 3.3.2 R Lie cebiri , M -bimodül olsun.

$$\mathrm{M} \times \mathrm{M} \to \mathrm{M}$$

$$(r_1, r_2) \to \left[m_1, m_2 \right] = 0$$

Bu durumda $0 : \mathrm{M} \to R$

$$x \to 0(x) = 0$$

sıfır morfizminin

$$R \times \mathrm{M} \to \mathrm{M}$$

$$(r, m) \to r \cdot m = rm$$

etkisi verilsin.

LieCM1: $0(r, m) = 0[r, m]$

$$= 0$$

$$= [r, 0m]$$

LieCM2: $\partial m_{m'} = 0m'$

$$= 0$$

$$= [m, m']$$

sağlandığından $(M, G, 0)$ bir çaprazlanmış modüldür.

Örnek 3.3.3 M bir Lie G-cebir, $\pi_2 : M \to G$ ile verilen ikinci izdüşüm fonksiyonu bir Lie G -cebir morfizmidir. $g' \in G, (g', m) \in G \times M$ için G 'nin $G \propto M$ üzerine Lie etkisi

$$g'(m, g) = (g'm, [g', g])$$

şeklinde tanımlanırsa $(G \times M, G, \pi_2)$ üçlüsü bir yarı çaprazlanmış modül olmasına rağmen genellikle bir çaprazlanmış modül değildir.

Tanım 3.3.4 (M, G, μ) ve (M', G', μ') Lie çaprazlanmış modüller,

$$f(g.m) = \phi(g).f(m)$$

ve $$\mu' f(m) = \phi\mu(m)$$

yani,

$$\begin{tikzcd} M \arrow[r, "f"] \arrow[d, "\mu"'] & M' \arrow[d, "\mu'"] \\ G \arrow[r, "\phi"'] & G' \end{tikzcd}$$

diagramı değişmeli olacak şekilde $f : M \to M' , \phi : G \to G'$ Lie **k-**cebir morfizmleri varsa,

$$(f,\phi) : (M, G, \mu) \to (M', G', \mu')$$

morfizmine izomorfizm denir. Bu durumda

$$(f,\phi)^{-1} = (f^{-1}, \phi^{-1}) : (M', G', \mu^{-1}) \to (M, G, \mu)$$

bir çaprazlanmış modül morfzmidir ve

$$(f,\phi)^{-1}(f,\phi) = (Id, Id) = (f, g)(f,\phi)^{-1}$$

dir. Böylece çaprazlanmış modüllerin bir kategorisi oluşturulur ve $LXMod(k)$ ile gösterilir. Özel olarak $G = G'$ ve ϕ birim dönüşüm ise f bir çaprazlanmış Lie G-modül morfizmidir. G üzerinde iki çaprazlanmış modülün bileşkesi de çaprazlanmış Lie G-modül morfizmi olduğundan $LXMod(k)$ nin alt kategorisi elde edilir ve $LXMod(k)/G$ ile gösterilir.

3.4. Cat^1-Lie Cebirleri

Cat^1- Lie cebiri ; $CL = (e; t, h : G \to P); t, h : G \to P$ Lie homomorfizmleri ve

LieCat1: $teh = h$, $het : t$

LieCat2: [ker t, ker h] = 0

ile tanımlanan $e : P \to G$ embeddinginden oluşur. Burada h, t başlangıç ve bitiş homomorfizmleri olarak tanımlanmıştır. $\partial : M \to P$ bir Lie homomorfizm ve $t(m, p) = p$ olmak üzere $h(m, p) = (\partial m) + p$ ve $e(p) = (0, p)$ yapısal morfizmleridir.

Lie cebirlerinin yarı direkt çarpımının tanımında der ki; M ve P modüllerinin çarpımı $M \times P$ A-modüldür. $\forall$ $\partial \in A$, $m, m_1 \in M$, $p, p_1 \in P$ için skaler çarpım ve toplam,

$$\partial(m,p) = (\partial m, \partial p)$$

$$(m,p) + (m_1, p_1) = (m + m_1, p + p_1)$$

ile verilir.

G Lie cebirinin yarı direkt çarpımı $G = M \times P = \{(m,p) \mid m \in M, p \in P\}$ ve

Lie braketi $\left[(m,p),(m_1,p_1)\right] = \left(\left[m,m_1\right] - {}^{p_1}m + {}^{p}m_1, \left[p,p_1\right]\right)$ ile tanımlanır.

Teorem 3.4.1 G Lie cebirinin oluşturduğu cat[1]-Lie cebiri, t, h yapısal homomorfizmleri ve e embeddingiyle tanımlanır.

İspat: LieCat1 ve LieCat2 koşullarının sağlandığını göstermeliyiz.

LieCat1: $teh = (m,p) = (te((\partial m) + p)$

$$= t(0, (\partial m) + p)$$

$$= (\partial m) + p$$

$$= h(m,p)$$

$$het(m,p) = he(p)$$

$$= h(0,p)$$

$$= (\partial 0) + p$$

$$= p$$

$$= t(m,p)$$

Böylece LieCat1 sağlanır. Şimdi LieCat2 nin sağlandığını gösterelim.

LieCat2: $x \in \ker t, y \in \ker h, x = (m,0), y = (m_1 - \partial m_1)$ için,

$$[x,y] = [(m,0),(m_1 - \partial m_1)]$$

$$=\left(\left[m,m_1\right]-{}^{-\partial m_1}m+{}^{0}m_1,\left[m,m_1\right]\right)$$

$$=\left(\left[m,m_1\right]-\left[-m_1,m\right],\left[m,m_1\right]\right)$$

$$=\left(mm_1-m_1m-(-m_1m-m(-m_1)),\left[m,m_1\right]\right)$$

$$=\left(mm_1-m_1m+m_1m-mm_1,\left[m,m_1\right]\right)$$

$$=\left(0,\left[m,m_1\right]\right)$$

dır. Böylece LieCat2 de sağlanır. Burada t ve h nin homomorfizm olduğu açıktır.

4. GERİ ÇEKME (PULLBACK) LİE CEBİR ÇAPRAZLANMIŞ MODÜL

Lie cebirinin aşağıdaki geri çekme diagramını göz önüne alalım.

$$\begin{tikzcd} & M \arrow[d, "\partial"] \\ Q \arrow[r, "\iota"'] & P \end{tikzcd} \quad \rightsquigarrow \quad \begin{tikzcd} \iota^{**}(M) \arrow[r, dotted] \arrow[d, dotted, "\partial^{**}"'] & M \arrow[d, "\partial"] \\ Q \arrow[r, "\iota"'] & P \end{tikzcd}$$

$\partial : M \to P$ Lie cebirlerinin çaprazlanmış P-modülü ve $\iota : \theta \to P$ Lie cebir homomorfizmi olsun. Buradan $\partial^{**} : \iota^{**}M \to \theta$, $\partial^{**}(q,m) = q$ Lie cebirlerin ι ile belirlenen geri çekme çaprazlanmış Q-modülüdür.

Önerme : Skaler çarpımı $\partial(q,m) = (\partial q, \partial m)$, toplamı $(q_1,m_1)+(q_2,m_2) = (q_1+q_2, m_1+m_2)$

ve $\left[(q_1,m_1),(q_2,m_2)\right] = \left(\left[q_{1,}q_2\right],\left[m_1,m_2\right]\right)$ ile verilen

$$\iota^{**}M = \{(q,m) \in \theta \times M \,|\, \iota q = \partial m, q \in Q, m \in M\}$$

kümesi bir Lie cebirdir.

Önerme : G nin $\iota^{**}M$ üzerindeki etkisi ${}^{q}(q_1,m) = (\left[q,q_1\right],\left[q,m\right])$ ile verilir.

İspat: İspatın tamamlanması için Lie cebiri için etki aksiyomlarının sağlandığını göstermeliyiz.

$$\text{Lie Act1: } {}^{(aq)}(q_1,m_1) = (\left[aq,q_1\right],\left[\iota aq,m_1\right])$$

$$= (a\left[q,q_1\right], a\left[\iota q,m_1\right])$$

$${}^{q}(a(q_1,m_1)) = {}^{q}(aq_{1,}am_1)$$

$$= (\left[q,aq_1\right],\left[\iota q,am_1\right])$$

$$= (a[q,q_1], a[\iota q, m_1])$$

$$= (a[q,q_1], a[\iota q, m_1])$$

Lie Act2:

$${}^{q}((q_1,m_1)) + (q_2,m_2)) = {}^{q}(q_1+q_2, m_1+m_2)$$

$$= ([q, q_1+q_2], [\iota q, m_1+m_2])$$

$${}^{q}(q_1,m_1) + {}^{q}(q_2,m_2) = ([q,q_1],[\iota q, m_1]) + ([q,q_2],[\iota q, m_2])$$

$$= ([q, q_1+q_2], [\iota q, m_1+m_2])$$

Lie Act3:

$${}^{(q_1+q_2)}(q,m) = ([q_1+q_2, q], [\iota(q_1+q_2), m])$$

$${}^{q_1}(q,m) + {}^{q_2}(q,m) = ([q_{1,}\, q], [\iota q_1, m]) + ([q_2, q], [\iota q_2, m])$$

$$= ([q_1+q_2, q], [\iota(q_1+q_2), m])$$

Lie Act4: ${}^{[q_1,q_2]}(q,m) = {}^{(q_1q_2 - q_2q_1)}(q,m)$

$$= ([[q_1,q_2],q], [\iota[q_1,q_2], m])$$

$$= ([q_1,[q_2,q]] - [q_2[q_1,q]], [\iota q_1,[\iota q_2, m]] - [\iota q_2, [\iota q_1, m]])$$

$$= ([q_1,[q_2,q]], [\iota q_{1,}\,[\iota q_2, m]]) - ([q_2,[q_1,q]], [\iota q_2, [\iota q_1, m]])$$

$$= {}^{q_1}({}^{q_2}(q,m)) - {}^{q_2}({}^{q_1}(q,m))$$

Buradan, ${}^{[q_1,q_2]}(q,m) = {}^{q_1}({}^{q_2}(q,m)) - {}^{q_2}({}^{q_1}(q,m))$ dir.

2. koordinat da kolayca gösterilebilir.

Lie Act5: ${}^{q}[(q_1,m_1),(q_2,m_2)] = {}^{q}([q_1,q_2],[m_1,m_2])$

$$= ([q,[q_1,q_2]], [\iota q, [m_1,m_2]])$$

$$=([[q,q_1],q_2],[[\imath q,m_1],m_2])+([q_1,[q,q_2]],[m_1,[\imath q,m_2]])$$

$$=\left[{}^{q}(q_1,m_1),(q_2,m_2)\right]+\left[(q_1,m_1),{}^{q}(q_2,m_2)\right]$$

Böylece 1.koordinat gösterilmiş olur.2.koordinatın sağladığı da kolayca gösterilebilir. Bu nedenle tanımlanan aksiyom, Lie aksiyomudur ve ispat tamamlansın.

Teorem : $\partial^{**}:\imath^{**}M\rightarrow\theta$ Lie homomorfizmi çaprazlanmış modül yapısına sahiptir.

İspat: $\partial^{**}(q,m)=q$ sınır homomorfizması ve θ nin M üzerindeki Lie cebiri etkisi LieCat1 ve LieCat2 koşullarını sağlar.

LieCat1: $\partial^{**}({}^{q}(q_1,m_1))=\partial^{**}([q,q_1],[\imath q,m_1])$

$$=[q,q_1]$$

$$=\left[q,\partial^{**}(q_1,m_1)\right]$$

LieCat2: $\partial^{**(q,m)}(q_1,m_1)={}^{q}(q_1,m_1)$

$$=([q,q_1],[\imath q,m_1])$$

$$=([q,q_1],[m,m_1])$$

$$[(q,m),(q_1,m_1)]=([q,q_1],[m,m_1])$$

Böylece çaprazlanmış modül aksiyomları sağlanır.

4.1. Geri Çekme (Pullback) Cat^1-Lie Cebir

$C=(e\,;\,t\,,\,s:A\rightarrow R)$ cat^1-lie cebir ve $\phi:S\rightarrow R,$ bir k - cebir morfizmi olsun.

$$\begin{tikzcd}
\phi^*(A) \arrow[r, "\phi'"] \arrow[d, shift right, "t^*"'] \arrow[d, shift left, "h^*"] & A \arrow[d, shift right, "t"'] \arrow[d, shift left, "h"] \\
S \arrow[r, "\phi"'] \arrow[u, bend left=60, "e^*"] & R \arrow[u, bend right=60, "e"']
\end{tikzcd}$$

değişmeli geri çekme diyagramında A nın $\phi^{*}(A)$ olan geri çekmesi olup, $\phi^{*}(C)$ ye **geri çekme cat^{1}-lie cebir** denir. Burada

$$\phi^{*}(C): (e^{*}\ ;\ t^{*}\ ,\ s^{*}\ :\ \phi^{*}(A)\ \rightarrow S)$$

olup

$$\phi^{*}(\mathrm{A}) = \{(s,a,s^{'}) | \phi(s) = t(a), \phi(s^{'}) = s(a)\} \subseteq S \propto A \times S$$

ve morfizmler

$$t^{*}\ (s\ ,a,\ s^{'}) = s$$

$$s^{*}\ (s\ ,a,\ s^{'}) = s^{'}$$

$$e^{*}\ (s) = (s,(e\phi)(s),s)$$

şeklinde tanımlanır.

Teorem 4.1.1

$$\phi^{*}(C): (e^{*}\ ;\ t^{*}\ ,\ s^{*}\ :\ \phi^{*}(A)\ \rightarrow S)$$

bir cat^{1} -lie cebiridir.

İspat:

CAT1)
$$\begin{aligned} t^{*}e^{*}(s) &= t^{*}(e^{*}(s)) \\ &= t^{*}(s,(e\phi)(s,s)) \\ &= s \\ &= 1_{s}(s) \end{aligned}$$

ve

$$\begin{aligned} s^{*}e^{*}(s) &= s^{*}(e^{*}(s)) \\ &= \mathrm{s}^{*}(s,(e\phi)(s),s) \\ &= s \end{aligned}$$

$$=1_s(s)$$

olup $t^*e^*(s)=1_s(s)=s^*e^*(s)$ dır.

CAT2) $x=(s_1',a_1,s_1)\in \ker t^*$ ve $x=(s_2,a_2,s_2')\in \ker s^*$ alalım.

$$\ker t^* \ker s^* = \{0\}$$

olduğunu göstermek için gerek ve yeter şart $xy=0$ dır.

$$xy=(s_1',a_1,s_1)\,(s_2,a_2,s_2')$$

$$=(s_1's_2a_1a_2s_1s_2')$$

dir. Fakat $x\in \ker s^*$ ve $y\in \ker s^*$ olduğundan

$$t^*(s_1',a_1,s_1)\;=s_1'=0_s$$

$$s^*(s_2,a_2,s_2')=s_2'\;=0_s$$

elde edilir.

Ayrıca $a_1\in Çekt$, $a_2\in Çeks$ ve C bir cat^1-lie cebir olduğundan

$$a_1a_2\in \ker t \ker s=\{0\}$$

olup $a_1a_2=0$ dır. Böylece

$$xy=((0_s),a_1a_2,0_s)=0$$

dır. Bu ise istenilendir. Bundan başka e^*,t^*,s^* nın homomorfizm oldukları kolayca gösterilir.

Önerme 4.1.2 (C,R,∂) çaprazlanmış R-modül ve $(\phi^*(C),S,\partial^*)$ geri çekmesi olsun. C ve D sırasıyla (C,R,∂) ve $(\phi^*(C),S,\partial^*)$ çaprazlanmış modüllerden elde edilen cat^1-lie değişmeli cebirler ise

$$D\cong \phi^*(C)$$

dir.

İspat: $\left(\phi^*(C), S, \partial^*\right)$ geri çekme çaprazlanmış S-modül olduğundan

$$\begin{tikzcd}
\phi^*(C) \arrow[r, "\phi'"] \arrow[d, "\partial^*"'] & C \arrow[d, "\partial"] \\
S \arrow[r, "\phi"'] & R
\end{tikzcd}$$

geri çekme diagramı vardır. Böylece

$$F : XMod \to Cat^1 - LieCeb$$

funktoru yardımıyla ∂ ve ∂^* çaprazlanmış modüllerine karşılık gelen cat^1-değişmeli cebirler sırasıyla

$$F(C,R,\partial) = \mathfrak{C} = \left(R \ltimes C \underset{e}{\overset{t,h}{\rightleftarrows}} R\right)$$

ve

$$F(\phi^*(C),S,\partial^*) = \mathfrak{D} = \left(S \ltimes \phi^*(C) \underset{e^*}{\overset{t^*,h^*}{\rightleftarrows}} S\right)$$

dır.

Ayrıca F funktoru yardımıyla geri çekmeleri taşıyabiliriz. Şöyle ki,

$$\begin{tikzcd}
S \ltimes \phi^*(C) \arrow[r, "\psi"] \arrow[d, shift right, "t^*"'] \arrow[d, shift left, "h^*"] & R \ltimes C \arrow[d, shift right, "t"'] \arrow[d, shift left, "h"] \\
S \arrow[r, "\phi"'] \arrow[u, bend left=60, "e^*"] & R \arrow[u, bend right=60, "e"']
\end{tikzcd}$$

olup D nin morfizmleri

$$t^*\left(s', (s,c)\right) = s'$$

$$s^*\left(s', (s,c)\right) = s' + \partial^*(s,c)$$

$$= s' + s$$

$$e^{*} = \left(s, \left(0_{s}, 0_{c}\right)\right)$$

şeklindeki homomorfizmlerdir. Şimdi D ve $\phi^{*}(C)\mathrm{cat}^{1}$-lie cebirlerinin birbirine izomorf olduğunu gösterelim. İlk olarak $\phi^{*}(C)$ geri çekme $\mathrm{cat}^{1}-$ cebirlerini oluşturalım. Yani,

$$\begin{tikzcd}
\phi^{*}(R \ltimes C) \arrow[r, "\phi'"] \arrow[d, shift right, "t^{\bullet}"'] \arrow[d, shift left, "h^{\bullet}"] & R \ltimes C \arrow[d, shift right, "t"'] \arrow[d, shift left, "h"] \\
\phi^{*}(R) = S \arrow[r, "\phi"'] \arrow[u, bend left=60, "e^{\bullet}"] & R \arrow[u, bend right=60, "e"']
\end{tikzcd}$$

$$\psi\left(s', (s, c)\right) = \left(s, \left(\phi\left(s'\right), c\right), s' + s\right)$$

tanımlayalım.

$$\phi^{*}(R \propto C) = \begin{Bmatrix} \phi(s) = t(r, c) \\ \phi(s) = s(r, c) \end{Bmatrix} \subseteq S \times (R \propto C) \times S$$

$r = \phi\left(s'\right)$ alınırsa $\psi\left(s', (s, c)\right) \in \phi^{*}(R \times C)$ olur. Çünkü,

$$t\left(\phi\left(s'\right), c\right) = \phi\left(s'\right)$$

ve

$$s\left(\phi\left(s'\right), c\right) = \phi\left(s'\right) + \partial(c)$$

$$= \phi\left(s'\right) + \phi(s)$$

$$= \phi\left(s' + s\right)$$

dir. ψ , S-lie cebir homomorfizmidir ve ψ nin tersi

$$\psi^{-1}\left(s_{1}, (r, c), s_{2}\right) = \left(s_{1}, \left(s_{2} - s_{1}\right), c\right)$$

olarak tanımlanır. Böylece,

$$\psi\psi^{-1}\left(s_1,(r,c),s_2\right)=\psi\left(s_1,(s_2-s_1),c\right)$$

$$=\left(s_1,\left(\phi(s_1),c\right),s_1+(s_2-s_1)\right)$$

$$=\left(s_1,\left(\phi(s_1),c\right),s_2\right)$$

$$=\left(s_1,(r,c),s_2\right)$$

$$=1_{\phi^*(R\propto C)}\left(s_1,(r,c),s_2\right)$$

Benzer şekilde,

$$\psi^{-1}\psi\left(s^{'},(s,c)\right)=\psi^{-1}\left(s^{'},\left(\phi\left(s^{'}\right),c\right),s^{'}+s\right)$$

$$=\left(s^{'},\left(s^{'}+s-s^{'},c\right)\right)$$

$$=\left(s^{'},(s,c)\right)\qquad\left(\because\phi^*(S)=R\right)$$

$$=1_{S\propto\phi^*(C)}\left(s^{'},(s,c)\right)$$

Ayrıca,

$$t^{\bullet}\psi\left(s^{'},(s,c)\right)=t^{\bullet}\left(s^{'},\left(\phi\left(s^{'}\right),c\right),s^{'}+s\right)$$

$$=s^{'}$$

$$=t^*\left(s^{'},(s,c)\right),$$

$$s^{\bullet}\psi\left(s^{'},(s,c)\right)=s^{\bullet}\left(s^{'},\left(\phi\left(s^{'}\right),c\right),s^{'}+s\right)$$

$$=s^{'}+s$$

$$=s^*\left(s^{'},(s,c)\right),$$

$$\psi e^*(s)=\psi\left(s,(0_S,0_C)\right)$$

$$=\left(s,\left(\phi(s),0_S\right),s\right)$$

$$= e^{\bullet}(s)$$

olup, diagram değişmelidir.

Şimdi geri çekme cat[1]-cebirin evrensellik özelliğine sahip olduğunu gösterelim.

Teorem 4.1.3 (Geri çekme cat[1]-lie cebir için evrensellik özelliği)

$C = (e; t, s : A \to R)$, $D = (e^{\bullet}; t^{\bullet}, s^{\bullet} : B \to S)$ cat[1]-cebirler olsun.

$$(f, \phi) : D \to C$$

cat[1]-cebir morfizmi olmak üzere,

$$\begin{tikzcd}
 & \mathfrak{D} = (e^{\bullet}; t^{\bullet}, h^{\bullet} : B \to S) \arrow[d, "(f,\phi)"] \\
(e^{*}; t^{*}, h^{*} : \phi^{*}(A) \to S) \arrow[ur, dashed, "(f^{*}, id_S)"] \arrow[r, "(\phi', \phi)"'] & \mathfrak{C} = (e; t, h : A \to R)
\end{tikzcd}$$

değişmeli olacak şekilde biricik (f^{*}, id_s) cat[1]-cebir morfizmi vardır. [18]

İspat: Burada $D = (e^{\bullet}; t^{\bullet}, s^{\bullet} : B \to S)$ herhangi cat[1]-cebir ve (f, ϕ) cat[1]-cebir morfizmi olmak üzere

$$f^{*} : \phi^{*}(A) \to B$$

biricik cat[1]-cebir morfizmi vardır, öyle ki $t^{\bullet} f^{*} = t^{*}$, $s^{\bullet} f^{*} = s^{*}$ ve $ff^{*} = \phi'$ dür.

İlk olarak (f, ϕ) çiftinin cat[1]-cebir olduğunu gösterelim.

Yukarıdaki diagramdan (Yani geri çekme cat[1]-cebir özelliğinden)

$$f : B \to A$$

$$(s, a, s') \to a$$

dir.

Buradan;

$$tf\left(s,a,s'\right)=t(a) \qquad\qquad sf\left(s,a,s'\right)=t(a)$$

$$=\phi(s) \qquad \text{ve} \qquad =\phi\left(s'\right)$$

$$=\phi t^{\bullet}\left(s,a,s'\right) \qquad\qquad =\phi s^{\bullet}\left(s,a,s'\right)$$

olup, (f,ϕ) çifti cat^1-cebir morfizmidir.

$$\left(\phi',\phi\right):C^{*}\rightarrow C$$

herhangi cat^1-cebir morfizmi öyle ki $t\phi'=\phi t^{*}$, $s\phi'=\phi s^{*}$ olmak üzere biricik

$$f^{*}:\phi^{*}(A)\rightarrow B$$

$$x\rightarrow\left(t^{*}(x),\phi'(x),s^{*}(x)\right)$$

morfizmi vardır, çünkü her $x\in\phi^{*}(A)$ için $t\phi'(x)=\phi t^{*}(x)$ ve $s\phi'(x)=\phi s^{*}(x)$ dir. imdi, f^{*} nin cat^1-cebir morfizmi olduğunu gösterelim. $x\in\phi^{*}(A)$, $s\in S$ için

$$\left(t^{\bullet}f^{*}\right)(x)=t^{\bullet}\left(f^{*}(x)\right)$$

$$=t^{\bullet}\left(t^{*}(x),\phi'(x),s^{*}(x)\right)$$

$$=t^{*}(x)$$

ve

$$\left(s^{\bullet}f^{*}\right)(x)=s^{\bullet}\left(f^{*}(x)\right)$$

$$=s^{\bullet}\left(t^{*}(x),\phi'(x),s^{*}(x)\right)$$

$$=s^{*}(x)$$

olup f^{*} cat^1-cebir morfizmidir. Ayrıca

$$\left(ff^{*}\right)(x)=f\left(f^{*}(x)\right)$$

$$=f\left(t^{*}(x),\phi^{'}(x),s^{*}(x)\right)$$

$$=\phi^{'}(x)$$

olup, $ff^{*}=\phi^{'}$ dir.

Teorem 4.1.4

$$\phi^{*}:Cat^{1}-Ceb/R\rightarrow Cat^{1}-Ceb/S$$

geri çekme funktoru vardır. [18]

İspat: $Cat^{1}-Ceb/R$ kategorisinde herhangi $C=\left(e;t,s:A\rightarrow R\right)$ cat^{1}-cebir objesi için geri çekme

$$\phi^{*}(C)=\left(e^{*};t^{*},s^{*}:\phi^{*}(A)\rightarrow S\right)$$

$Cat^{1}-Ceb/S$ kategorisinde bir objedir.

Örnek 4.1.5 $\mathrm{I}\trianglelefteq R$ olsun. Bu durumda $\partial:\mathrm{I}\rightarrow R$ içine çaprazlanmış R-modülünün $\phi:S\rightarrow R$, k-cebir homomorfizmi yardımıyla elde edilen geri çekme çaprazlanmış modülünün

$$\phi^{*}(\mathrm{I},R,\partial)=\left(\phi^{*}(\mathrm{I}),S,\partial^{*}\right)$$

$$\cong\left(\phi^{-1}(\mathrm{I}),S,\partial^{*}\right)$$

ve

$$\phi^{*}(\mathrm{I})\cong\phi^{-1}(\mathrm{I})\trianglelefteq S$$

olduğu gösterilmişti. Ayrıca önermeden,

$$\mathfrak{C}=\Big(R\ltimes I\;\underset{\xleftarrow[e]{}}{\overset{t,h}{\rightrightarrows}}\;R\Big)$$

ve

$$
\begin{aligned}
\mathfrak{D} \cong \quad \phi^{*}(\mathfrak{C}) &= \left(\phi^{*}(R \ltimes I) \mathrel{\substack{\longrightarrow\\ \longrightarrow\\ \longleftarrow}} \phi^{*}(R)\right) \\
&= \left(S \ltimes \phi^{*}(I) \mathrel{\substack{\longrightarrow\\ \longrightarrow\\ \longleftarrow}} S\right) \\
&= \left(S \ltimes \phi^{-1}(I) \mathrel{\substack{\longrightarrow\\ \longrightarrow\\ \longleftarrow}} S\right)
\end{aligned}
$$

dir.

KAYNAKLAR DİZİNİ

[1] Alp, M., 1997, GAP, Crossed Modules, Cat^1-groups Applicatons of computational Group Theory, Ph.D Thesis University of Wales, Bangor.

[2] Alp, M., 1998, Pullbacks of Crossed Modules and Cat^1-groups, Turkish Journal of Mathematics, Vol. 2, No. 3, 273-281.

[3] Alp, M., 1999, Left Adjoint of Pullback Cat^1- groups, Turkish Journal of Mathematics, Vol. 23, No. 2, 243-249

[4] Alp, M., 2001, Induced Cat^1-groups, Turkish Journal of Mathematics, Vol. 25, No. 2, 245- 261.

[5] Alp, M., 2002, Enumeration Of Whitehead groups Of Low Order, International Journal of Algebra and Computation, Vol. 12, No. 5, 645-658.

[6] Alp, M., 2002, Left Adjoint of Pullback Cat^1-profinite groups, Turkish Journal of Mathematics, Vol. 26, No. 2, 159- 167.

[7] Alp, M. and Wensley, C. D., 1997, Crossed Modules and Cat^1-groups in GAP, Version 1-3 Manual for the XMOD share package, Chapter 73, 1357-1422.

[8] Alp, M. and Wensley, C. D., 2000, Enumeration Of Cat^1-groups Of Low Order, International Journal of Algebra and Computation, Vol. 10, No. 4, 407-424.

[9] Aytekin, A, 2005, Lie Cebirlerin Çaprazlanmış Modülleri, Dumlupınar Üniversitesi Yüksek Lisans tezi.

[10] Brown, R. and Higgins, P. J., 1978, On the connection between the second relative homotopy groups of some related spaces, Proc. Londan, Math. Soc., 36, 193-212.

[11] Brown, R. and Loday, J. L., 1987, Homotopical Excision and Hurewicz Theorems for n-cubes of Spaces, Proc. London, Math. Soc., (3)54, 176-192.

[12] Brown, R. and Wensley, C.D., 1995, On Finite Induced Crossed Modules and The Homotopy 2-Type Of Mapping Cones, Theory and App. Of Categories, Vol. 1, No. 3, 54-71.

KAYNAKLAR DİZİNİ (devam)

[13] Bovdi,V., Konovalov, A., Rossmanith, R., ve Schneider, C., 2007, LAGUNA, Lie Algebras and UNits of group Algebras, GAP ortak paketi.

[14] Casas, J.M., ve Ladra, M., 1998, The actor of a crossed module in Lie Algebras, 26 (7), 2065-2089.

[15] Casas, J.M., Ladra, M., 2000, Colimits in the CrossedModules Category in Lie Algebras, GeorgianMathematical Journal, 7, 3, 461-474.

[16] Ellis, G.J., 1984, Crossed Modules and their higher dimensional analogues, Ph.D Thesis University of Wales, Bangor.

[17] Gürmen, Ö., 2003, Profinite Çaprazlanmış Modüller, Dumlupınar Üniversitesi Yüksek Lisans Tezi.

[18] Gürmen, Ö., 2007, Geri Çekme, İleri İtme Çaprazlanmış Modüller, Cat^1-Cebirler ve Simplisel Cebirler, Eskişehir Osmangazi Üniversitesi Doktora Lisans Tezi.

[19] Judson, T. W., 1993, Abstract Algebra Theory and Applications, PWS Publishing Company, Boston.

[20] Loday, J. L, 1982, Spaces with finitely many non-trivial homotopy groups, J. App. Algebra, 24, 179-202.

[21] Schönert, M. ET AL, 1993, GAP: Groups, Algorihms and Programming, Lehrstulh D für Mathematik, Rheinisch Westfalishe Technische Hochschule, Aachen, Germany, third Edition.

[22] Whitehead, J.H.C., 1948, On operators in relative homotopy group, Ann. Of Math, 49, 610-640.

[23] Whitehead, J.H.C., 1949, Combinatorial homotopy II, Bull. A. M. S. 55, 453-496.