

自律システム相互接続点における経路制御運用の自動化のための技術要素の提案

衛藤 将史 森島 直人 門林 雄基

† 奈良先端科学技術大学院大学情報科学研究科
〒 630-0101 奈良県生駒市高山町 8916-5

E-mail: {masash-e, naoto-mo, youki-k}@is.aist-nara.ac.jp

あらまし 自律システム相互接続点における経路制御運用の支援技術として RPSL と呼ばれる言語が開発されている。IRR(Internet Routing Registry) と呼ばれるサーバはこの RPSL によって記述された経路制御ポリシーを管理するためのものであるが実運用においてはほとんど利用されていないのが現状である。その原因の 1 つとして各 AS(Autonomous System) の管理者が他の AS とのポリシーの整合性を検査することなく、それぞれ独自のポリシーを登録していることが挙げられる。ポリシー間に生じた不整合は予期せぬ経路選択を招くことになる。本論文ではこのポリシー間の不整合を検出し管理者に対して警告を発するシステムを提案し、そのアルゴリズムの提示を行なう。

キーワード AS, IX, 経路制御運用, RPSL

A Component of Automation System for Routing Operation in Autonomous Systems Exchange Point

Masaki ETO, Naoto MORISHIMA, and Youki KADOBAYASHI

† Nara Institute of Science and Technology
Graduate School of Information Science

Abstract As an automation system of routing operation in exchange point of the Autonomous System, RPSL (Routing Policy Specification Language) and IRR (Internet Routing Registry) system have been developed. Although IRR has been designed to manage routing policies described in RPSL, this system is not extensively used in many routing operations. The maintainers of ASes are registering their own routing policies without checking consistency between the policies of other ASes. As a result, this prevents increasing the utilization of IRR. Inconsistency between routing policies results in unexpected routing. In this paper, we propose a system which detects these inconsistencies and notifies the maintainer to correct them. The algorithm for detecting inconsistencies are presented.

Key words AS, IX, Routing Operation, RPSL

1. はじめに

インターネットは、自律システム (Autonomous System: AS) と呼ばれる独立したネットワークが相互に接続された形態で構成されている。各 AS 毎に提供されるサービスが異なる為、各々が固有のポリシーに則って運営されている。

AS 同士の接続に際しては主に 2 つの形態がある。2 つの AS が専用回線を用いた Private Peering で経路情報を交換する形態と、複数の AS が相互接続点 (IX) において回線を共有し経路情報を交換する形態とである。これら AS 間の経路制御には BGP(Border Gateway Protocol) などの外部経路制御プロトコルが用いられている。

その一方で、このようなネットワークにおける経路制御運用の多くの部分は、人手によって行われている。人間による運用は人的コストと、人為的な設定ミスによるトラフィック障害のリスクを負うことになる。また、ルータのベンダによってルータ設定情報の記述仕様や機能が異なる点も運用者の負担を増加させる原因となっている。

このような問題に対処する方法として RPSL(Routing Policy Specification Language) と呼ばれるポリシー記述言語が開発されている。RPSL は主として AS 間のポリシーを表現する為に設計され、このポリシーから RtConfig と呼ばれるツールを用いることでルータのルータ設定情報を自動生成することが出来る。このような機能により経路制御運用の自動化が図られているが、

実環境においてこれらの技術が普及しているとは言い難い。

本研究ではこの点に着目した。経路制御運用の自動化の観点から RPSL の問題点を指摘し、その問題を克服するための技術を提案することで、経路制御運用の自動化の実現への足掛かりにしたいと考えている。

本稿では 2 章にて RPSL, RtConfig といった経路制御運用の自動化技術について説明する。3 章ではそれらの抱える問題点を指摘し、4 章で本研究における提案技術について幾つかの実例を挙げながら解説する。最後にまとめと考察を述べる。

2. 既存の経路制御運用支援技術

本章においては既存の経路制御運用支援技術である RPSL と RtConfig について簡単にまとめる。RPSL は主として AS 間のポリシーを表現する為に設計されたポリシー記述言語である。BGP ネットワークでは個々の接続点において隣接する AS ごとに、経路の告知、経路の集約、パス属性、経路の受け入れをどのようにするかを決めることができる。RPSL ではこれらのポリシーをより運用者にわかりやすい形で記述することが出来る。記述されたポリシーは、IRR(Internet Routing Registry) と呼ばれるホストのデータベースに対して登録される。登録されたデータは必要に応じて外部に公開され、適宜取り出すことが出来る。

RtConfig は上述のデータベースに対して問い合わせを行い、ルータ設定情報のうちポリシーを実現する部分を生成するためのツールである。RtConfig は RPSL から Cisco, Rsd, Nortel Bay といったベンダのルータや Gated の設定ファイルを生成することが出来る。

このようにルータベンダに非依存のポリシー記述言語から各ルータ固有のルータ設定情報を生成することで、様々なベンダのルータへの対応と、ルータ設定情報の記述の際のミスを防ぐことが出来る。

3. 問題点

しかし実環境においてこのシステムが利用されている例は少ない。その原因は RPSL を取り巻くシステムの運用形態にあると言える。現在、RPSL によって記述されたポリシーは IRR にて登録・公開されている。各 AS はこの IRR に対して別個独立して独自のポリシールールを登録している。登録に際しては irrd と呼ばれるデーモンプログラムが、RPSL で記述されたポリシーについて文法的な検査を行っている。しかしこの時にポリシーの論理的な整合性の検査までは行われていない。また、RtConfig は各 AS の記述したポリシーデータのみを元にルータ設定情報を生成している為、当該 AS と隣接関係にある AS とのポリシーの矛盾などを検出することは出来ない。

このシステムによると、仮にある 2 つの AS 間で契約上定められたポリシーとは矛盾するポリシーが登録された場合、RtConfig は 2 つの AS のルータに対して矛盾するルータ設定を生成することになる。これにより意図されない経路情報が誤って広告されてしまった場合、通信に支障を来すおそれがある。インターネット上に流れている経路情報が増加し続けている中、誤情報が流れたときの影響は深刻である。このように、IRR に登録さ

れたデータベースにおける経路情報は常に整合性が保たれた状態になければならない。

では具体的にポリシー間の不整合とはどのような場合を指すのであろうか。以下に問題となる不整合の例を挙げて説明する。

3.1 要求する経路の不到達

AS65534 と AS65533 は隣接関係にある。図 1 はこれを図示している。契約上は AS65533 が AS65534 に対して transit となり AS65532, AS65531, AS65530 への経路を広告し、それを AS65534 が受け取ることとなっているとする。しかし実際に登録されているポリシーは図 2,3 の通りである。

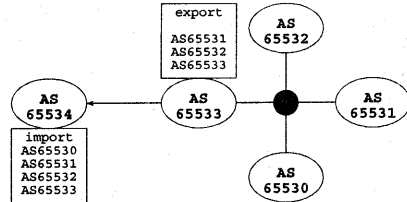


図 1 要求する経路の不到達

```
aut-num: AS65534
as-name: EtoNet
...
import: from AS65533
        accept AS65532,AS65531,AS65530
...
```

図 2 AS65534 が登録したポリシー情報

```
aut-num: AS65533
as-name: SaiNet
...
export: to AS65534
        announce AS65533,AS65532,AS65531
...
```

図 3 AS65533 が登録したポリシー情報

これらの図は AS65534 と AS65533 のそれぞれが IRR に対して登録したポリシー情報である。ここで、それぞれのポリシーにおける import および export 行に注目すると、AS65534 は他の AS への経路を AS65533 を経由して受け取る旨の宣言をしている。一方、AS65533 は AS65534 に対して自らの経路と AS65532 及び AS65531 への経路のみを広告しよう宣言している。これらのポリシーをもとに生成された gated のルータ設定情報は図 4,5 のようになる。

これらを比較すると AS65534 内ルータにおけるルータ設定情報では AS65530 の経路を受け取る旨の記述がなされているが、AS65533 内ルータにおいては AS65530 の経路は広告されていないことがわかる。前述のとおり AS65534 及び AS65533 の間では AS65533 が AS65534 に対して AS65532,AS65531,AS65530

```
import proto bgp as AS65533 {
  192.168.2.0 masklen 24 exact; /AS65533 の経路/
  192.168.3.0 masklen 24 exact; /AS65532 の経路/
  192.168.4.0 masklen 24 exact; /AS65531 の経路/
  192.168.5.0 masklen 24 exact; /AS65530 の経路/
  all restrict;
};
```

図 4 AS65534 内ルータにおけるルータ設定情報

```
proto bgp aspath .* origin any {
  192.168.2.0 masklen 24 exact; /AS65533 の経路/
  192.168.3.0 masklen 24 exact; /AS65532 の経路/
  192.168.4.0 masklen 24 exact; /AS65531 の経路/
  all restrict;
};
```

図 5 AS65533 内ルータにおけるルータ設定情報

に対する transit となるとの契約が結ばれている。これらのルータ設定情報がそのままルータに反映された場合、契約通りに意図されたルーティングは行われなかったことになり、問題となる。

3.2 送出した経路の不到達

前述の例とは逆に契約通りに送出された経路が受信側によって受け取られない場合がある。同様の例で図 6 を元に確認を行う。

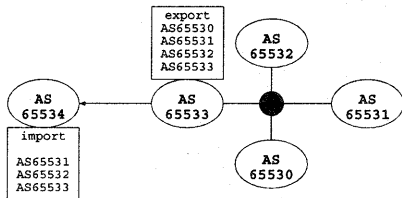


図 6 送出した経路の不到達

```
aut-num: AS65534
as-name: EtoNet
...
import: from AS65533
       accept AS65533,AS65532,AS65531
...
```

図 7 AS65534 が登録したポリシー情報

```
aut-num: AS65533
as-name: SaiNet
...
export: to AS65534
       announce AS65533,AS65532,AS65531,AS65530
...
```

図 8 AS65533 が登録したポリシー情報

図 7 の AS65533 について登録されているポリシー情報によると AS65533 は自らの経路と、AS65532,AS65531,AS65530 への経路を広告しようと言っている。一方、図 8 の AS65534 について登録されているポリシー情報によると AS65533,AS65532,AS65531 への経路しか受信される旨の記述がされていない。これらのポリシーから RtConfig を用いてルータ設定情報を生成した例は図 9,10 の通りである。

```
import proto bgp as AS65533 {
  192.168.2.0 masklen 24 exact; /AS65533 の経路/
  192.168.3.0 masklen 24 exact; /AS65532 の経路/
  192.168.4.0 masklen 24 exact; /AS65531 の経路/
  all restrict;
};
```

図 9 AS65534 内ルータにおけるルータ設定情報

```
proto bgp aspath .* origin any {
  192.168.2.0 masklen 24 exact; /AS65533 の経路/
  192.168.3.0 masklen 24 exact; /AS65532 の経路/
  192.168.4.0 masklen 24 exact; /AS65531 の経路/
  192.168.5.0 masklen 24 exact; /AS65530 の経路/
  all restrict;
};
```

図 10 AS65533 内ルータにおけるルータ設定情報

経路情報を受ける側の AS65534 において、AS65533 が広告する AS65530 に関する経路情報は all restrict 文によって破棄される。契約上は AS65530 への経路も広告することになっているため、AS65534 におけるポリシー情報は正されなければならない。仮に契約が AS65533,AS65532,AS65531 への経路のみをやりとすればよいとの内容であった場合は AS65533 は不要な経路を広告していることになる為、この場合もその記述を正す必要がある。

3.3 med, pref の指定

RPSL では med(Multi-Exit-Discriminator) と pref(Local-Preference) の 2 つの属性を指定することができる。仮に隣接関係にある 2 つの AS が互いに 2 つのルータを構成し接続している場合、その 2 つの接続は冗長経路となり優先度の指定をすることができる。この際に使用されるのが med と pref 属性である。経路の優先度を指定するとき、経路情報の送出側は med 属性を利用して優先させたい経路にもう 1 つの経路よりも低い値を指定する。一方、経路情報の受信側は pref 属性を用いて受け取った経路情報に優先する経路指定を付加する。

この際考慮すべき点は経路情報の送出側と受信側で優先させる経路の指定が異なる場合である。送出側において med 属性で優先指定された経路に対し、受信側の pref 属性によってより低い優先度が与えられた場合はポリシーの不整合として扱うべきであろうか。

この点、BGP においては経路情報に med,pref 属性の指定があった場合、pref 属性をもって med 属性よりも優位性をもた

せるとの指定がある (RFC1773)。よってシステム運用者によって明示的に pref 属性による優先経路の指定がなされている場合は一概にポリシの不整合とは判断し難いものがある。

ただし、図 11 に示すような冗長経路において med と pref 属性によって非対称経路を構成している場合も考えられる。パケットの往路と復路が異なる様に構成される非対称経路においてはあらかじめ設計されたとおりの優先経路指定がなされていない場合にはパケットが意図とは違った経路を流れる可能性がある。よって経路情報の送出側と受信側において複数の経路に矛盾する優先指定を与えている場合はなんらかの対処が必要である。

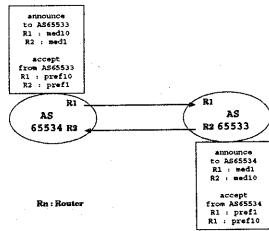


図 11 非対称経路の構成

4. 提案技術

4.1 問題の解決へのアプローチ

これらの問題の根底には AS が別個独立にポリシの登録を行っていることが原因として考えられる。現在の運用において、ポリシの登録の際に検査されるのは RPSL の文法的な整合性のみである。しかし IRR に登録されるポリシは AS 同士の経路情報のやり取りを記述するものであるから、その登録の際にポリシ間の整合性についても検査し、矛盾の無いようにすべきである。

そこで本研究ではポリシの登録の際にある一定の規則に従って AS 間のポリシの整合性を検査するシステムを提案する。

本システムでは図 12 の様な構成で PolicyCheck サーバを設置する。現状において各 AS はポリシ登録の際に IRR に接続し、ポリシの登録を行う。本システムでは、その前段階として各 AS が PolicyCheck サーバを用いてポリシの整合性検査を行う。整合性の検査はデマンド方式で行われ、仮に他に登録されたポリシとの不整合が検出された場合にはエラーまたは警告が通知される。報告を受けた登録者は該当部分を修正した上で IRR に登録を行うという手順を踏む。また PolicyCheck サーバは IRR サーバにあるポリシデータベースの全エントリを複製・保存し、最新のデータベースにて整合性検査を行う。

4.2 整合性検査の範囲

では実際に、整合性の検査はどのような範囲において行うべきであろうか。AS は主に IX にて相互接続を行い、経路情報のやり取りを行っている。そこで、第一に IX を中心とした整合性検査が考えられる。

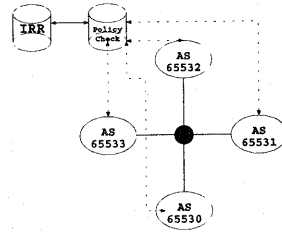


図 12 基本構成

4.2.1 IX を中心とした整合性検査

前述の通り、AS は主に IX にて相互接続を行っている。そこで IX を中心とした整合性検査では 1 つの IX において形成される隣接関係を網羅して整合性の検査を行う。そのため PolicyCheck サーバを各 IX 上に構成する。例えば、図 12 の場面において AS65534 に関するポリシを登録する場合には、IX 上の PolicyCheck サーバは AS65534 と隣接関係をつ結んでいる AS65533, AS65532, AS65531 のポリシを検査する。

しかしこれでは図 13 のような場面において不都合が生じることが考えられる。

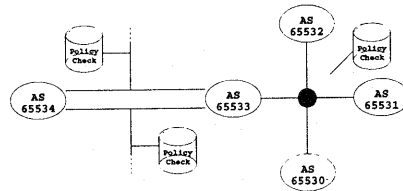


図 13 IX をまたがったシステム構成

図 13 のように 2 つの AS が異なる 2 つの IX において隣接関係を結んでいる場合、ここで述べたような IX を中心とした整合性検査の手法では精密な検査を行えないことがある。例えば AS65534 から AS65533 へ向かう経路については IXa にて行い、逆に AS65533 から AS65534 へ向かう経路については IXb にて行うといった場合である。このような場合、IXa における AS65534, AS65533 間のポリシ記述と IXb におけるそれとを比較する術が無い為、両者間の不整合を検出できない可能性がある。よって本研究においては IX を中心とした整合性検査とは異なる手法を採用することにする。

4.2.2 隣接 AS 間のポリシにもとづく整合性検査

インターネットは主に AS によって構成されている。それらを相互に接続するのは IX だけでなく専用線を用いた private peering といった形態も存在する。AS によっては private peering と IX による接続を同時に行っているものもあり、一概に IX を中心として整合性を検査するには不都合な場合があるというのは前述の通りである。そこで本研究では AS を中心とした検査方法を採用する。

本手法では、まず PolicyCheck サーバを任意の場所に設置し、IRR に登録されている全データベースを複製して検査に備

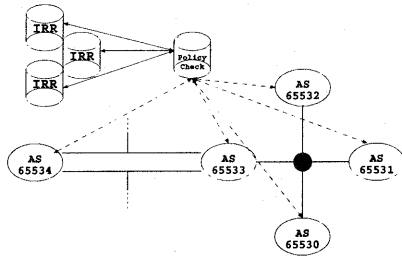


図 14 隣接関係にもとづく整合性検査

える。図 14 の例では AS65534 に関するポリシーを登録する際に全データベースのエントリの中から AS65534 と隣接関係を結んでいる全 AS に関するポリシーを抽出する。これら抽出されたポリシーを比較することで AS65534 にまつわる全ての AS のポリシーを検査することが出来る。

4.3 整合性検査のアルゴリズム

では、実際にポリシーの整合性を検査するアルゴリズムを図 15、16 に示す。経路の到達性に関する検査については図 15 に示されている。また図 16 では非対称経路におけるポリシーの検査アルゴリズムを示している。実際にはこれらのアルゴリズムの両方を用いて検査を行うことになる。

1. 隣接 AS の特定

- 入力された aut-num オブジェクトの import, export 行を抽出する
- 残された行の from, to で指定された AS を隣接 AS として確保する

2. import 行の検査

- 確保された入力行の import 文の数だけ以下の処理を繰り返す
 - = 隣接 AS の export 文の数だけ以下の処理を繰り返す
 - もし export 文の to 指定が自 AS であれば
 - 入力エントリの accept 文と隣接 AS エントリの announce 文を比較する
 - もし accept 文の AS 番号が announce 文内になければ
 - 要求した経路が広告されていないのでエラー出力
 - もし accept 文にない AS 番号が announce 文にあれば
 - 要求されていない経路を広告しているので警告を出力

3. export 行の検査

- 確保された入力行の export 文の数だけ以下の処理を繰り返す
 - = 隣接 AS の import 文の数だけ以下の処理を繰り返す
 - もし import 文の from 指定が自 AS であれば
 - 入力エントリの announce 文と隣接 AS エントリの accept 文を比較する
 - もし announce 文の AS 番号が accept 文内になければ
 - 要求されていない経路を広告しているので警告を出力
 - もし announce 文にない AS 番号が accept 文にあれば
 - 要求した経路が広告されていないのでエラー出力

図 15 経路広告の検査アルゴリズム

4.4 IRR 未登録の AS への対応

RPSL と IRR の段階的な普及の為には IRR に未登録の AS についてもその存在を許容できるシステムでなければいけない。この点に関しては隣接関係を結んでいる AS に関わるエントリ

が存在しなかった場合には、その旨の警告を発生し IRR への登録を促すのみとする。また 1 つの AS に関するエントリが存在しないと、それにまつわる多くの警告が発生されることが予想される。この点に関しては 1AS のエントリの存否により警告を集約して通知する仕組みが必要であると考えられる。

5. 考 察

経路制御運用の自動化については今回指摘した点以外にも解決すべき課題は多い。その例は特に IRR サーバの運用形態について挙げられる。現在の IRR サーバは ISP が独自に立ち上げているものが多く、その構成についての規則が存在しない。よって IRR サーバは乱立状態となっており、1 つのサーバで確実に全ての経路情報を保持しているとは言い難い状況が続いている。この点に関しては JPNIC の IRR 研究会においても指摘されてはいるが明確な解決策が現れていないのが現状である。

今回提案する PolicyCheck サーバは基本的には IRR に登録されたポリシーデータベースを複製することでポリシーの整合性を検査する手法を採用している。このため、複製を行うサーバによっては不十分な経路情報しか登録されていないことが考えられ、整合性の検査に支障を来すことも予想される。この点については PolicyCheck サーバが複数の IRR サーバからデータベースを複製することで、より確実なデータを保持することが出来るのではないかと考えている。

しかしより確実に期す為には、やはり IRR サーバの構成に一定の規定を課すべきではないかと考えている。例えば DNS のような階層構造を設け、登録情報が自動的に伝播するような仕組みを導入すれば、より確実で使いやすいものと思われる。

6. 関連研究

IRR に登録されているポリシーの内容一貫性に関しては RIPE の RRCC(Routing Registry Consistency Check) プロジェクトによって研究が進められている。RRCC の Web サイトでは IRR のデータベースに登録されていないがインターネットに広告されていない経路や、逆にインターネットに広告されていないが IRR への登録がされていない経路を検出するためのツールが公開されている。RRCC において行われている研究は登録された IRR データベースの内容と現実インターネットで広告されている内容との比較および不整合の検出である。この点、本研究ではともに登録されているポリシー同士において不整合の検出を行なうものである。RRCC のツールとは異なる側面でポリシーデータベースの不整合を正すことで、より安定した経路情報のやりとりを行うことができると考えている。

7. ま と め

本稿では IRR と RPSL 及び RtConfig を用いた IX における経路制御運用の自動化に関して、その問題点を指摘し、一解決策を提示した。これらの技術においてはポリシーに関する文法的な検査は行うが AS 間のポリシーに関する整合性の検査については行われていない。これにより隣接 AS 間のポリシーに不整合が生じる場合があることを指摘した。

ここでいう不整合とは大きく分けて3つ挙げられる。1つ目として隣接 AS 間において一方の AS が要求する経路情報をもう一方の AS から広告されない点。逆に一方の AS が広告した経路情報がもう一方によって受信されないというのが2点目である。3つ目は隣接 AS 間において med,pref といった経路選択のパラメタの指定に矛盾があることにより、予期しない経路選択につながるという点である。

これらの矛盾を解消する手法として、本稿ではポリシー間の不整合を検出するシステムの提案を行った。このシステムは IRR サーバにポリシーを登録する前段階において機能する。登録される AS オブジェクトに関連する他の AS のポリシーをポリシーデータベースより洗いだし、1つ1つのエントリについて不整合を検査し、不整合が認められる場合は登録者に対して警告を発するというものである。

さらに、不整合の検査についてそのアルゴリズムを提示した。

文 献

- [1] 堀良彰、池永全志、門林雄基、後藤滋樹、"ネットワークの相互接続" 岩波書店、2001.
- [2] C.Alaettinoglu, "Routing Policy Specification Language(RPSL)" RFC2280, 1998. <http://www.ietf.org/rfc/rfc2280.txt>
- [3] P.Traina, "Experience with the BGP-4 protocol" RFC1773, 1995. <http://www.ietf.org/rfc/rfc1773.txt>
- [4] IRR 研究会、"IRR に関する検討報告" 社団法人日本ネットワークインフォメーションセンター、2000
- [5] RIPE, "Routing Registry Consistency Check" <http://www.ripe.net/ripence/pub-services/db/rrcc/>

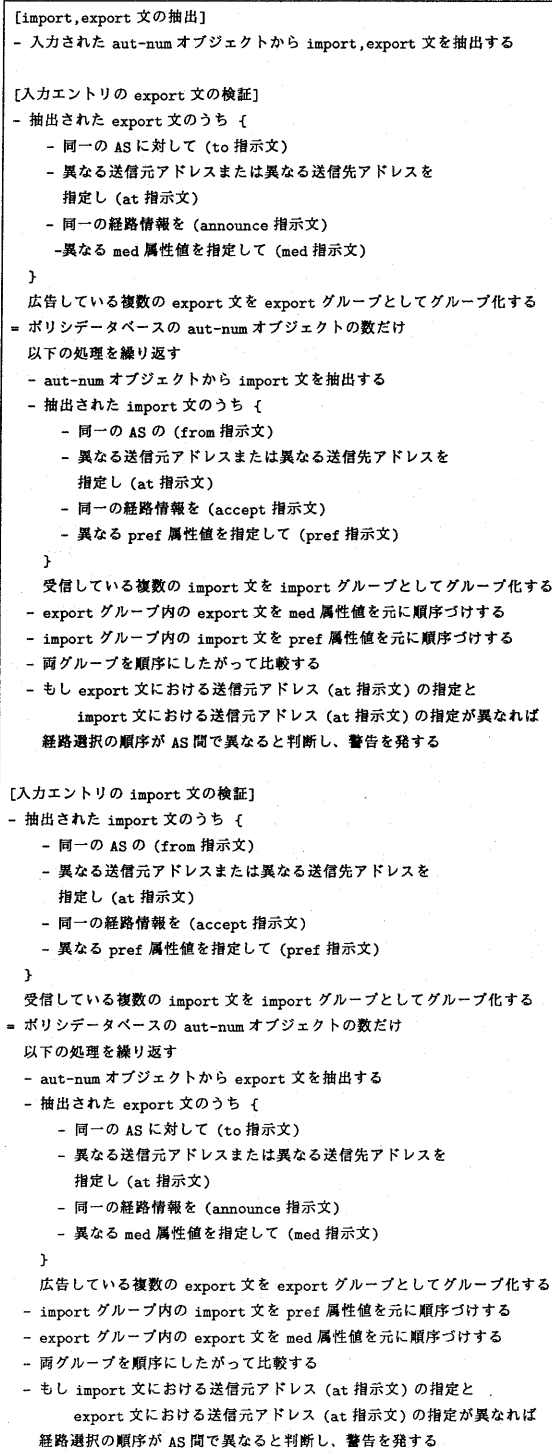


図 16 非対称経路におけるポリシー不整合の検査アルゴリズム