Nish!gaki
Laboratory

自動実行登録に基づくマルウェアの分類に関する検討と評価

静岡大学　名坂　康平
静岡大学　酒井　崇裕
静岡大学　山本　　匠
KDDI研究所　竹森　敬祐
静岡大学　西垣　正勝

マルウェアの検知技術

Nish!gaki Laboratory

- パターンマッチング法
  - マルウェアのバイトパターンを定義し、マッチングすることで検知する
- ビヘイビアブロッキング法
  - マルウェアの振る舞いを定義し、その振る舞いを行っているかを監視することで検知する

パターンマッチング法

- 既知のマルウェアに対して有効

しかし

- マルウェアの新種・亜種の生成速度の増加
  - マルウェア作成ツールの出現
  - 暗号化・難読化

そのため

- 定義ファイルの更新が間に合わない
- 未知のマルウェアに対応できない

ビヘイビアブロッキング法

Nish!gaki Laboratory

- マルウェア特有の挙動を監視することで，未知のマルウェアに対しても有効

しかし

- マルウェア特有の挙動を定義することが難しい

既存方式（始めに広くチェックする方式）

Nish!gaki Laboratory

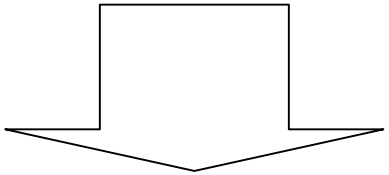
「Windows APIの監視による不正インストール検出手法の提案」

- レジストリ関連のWindowsAPIを監視し，使われた際にアラートを表示する.

既存方式（始めに広くチェックする方式）

Nish!gaki Laboratory

- 監視対象のレジストリを操作するAPI
  - マルウェアだけでなく，一部の正規プログラムも利用する



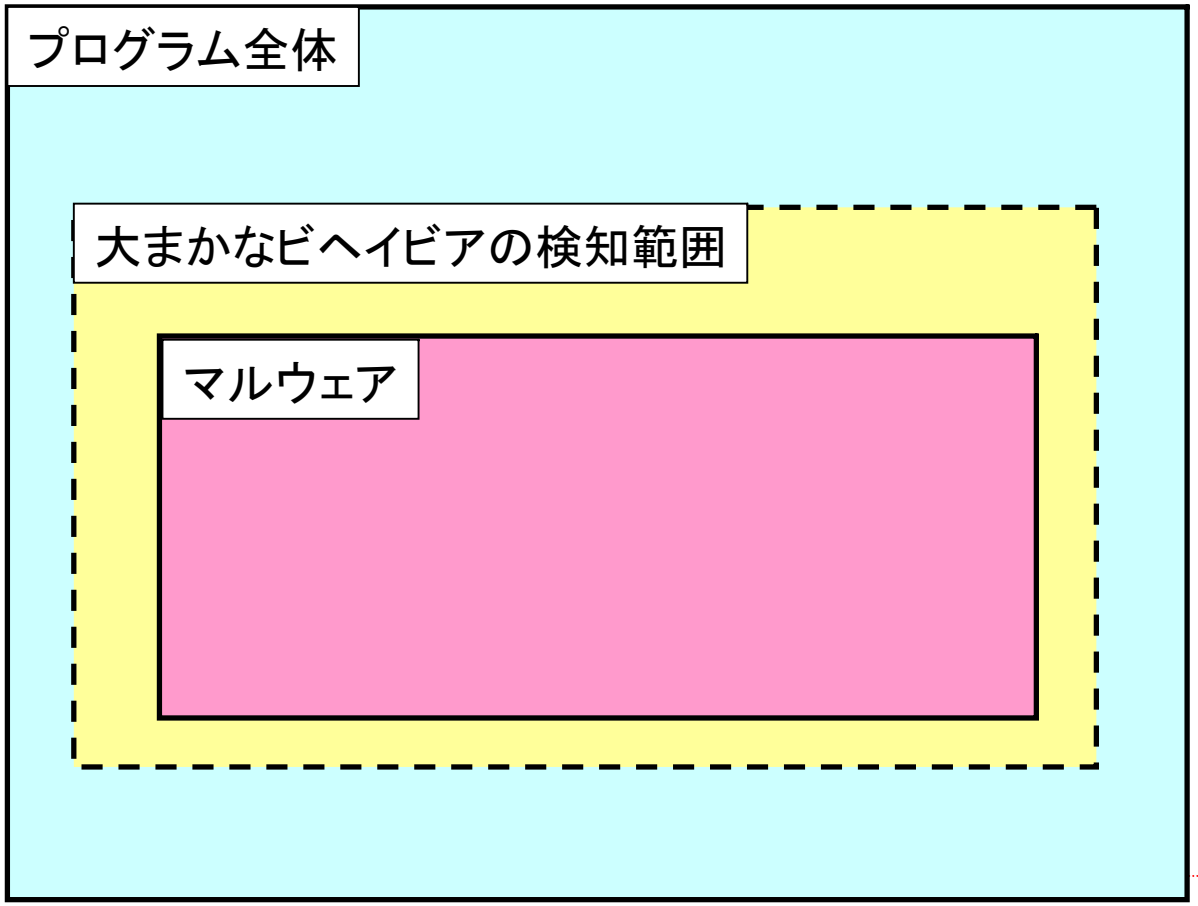
誤検知の原因

- ユーザに判断させることで最終的な処理を決定
  - ユーザビリティの低下
  - ユーザの誤判断

ビヘイビアを定義する際の課題

Nish!gaki Laboratory

大まかなビヘイビアでは正規プログラムも誤検知してしまう



マルウェアの
本質を捉えた挙動
の発見が必要

既存方式（狭くチェックする方式）

Nish!gaki Laboratory

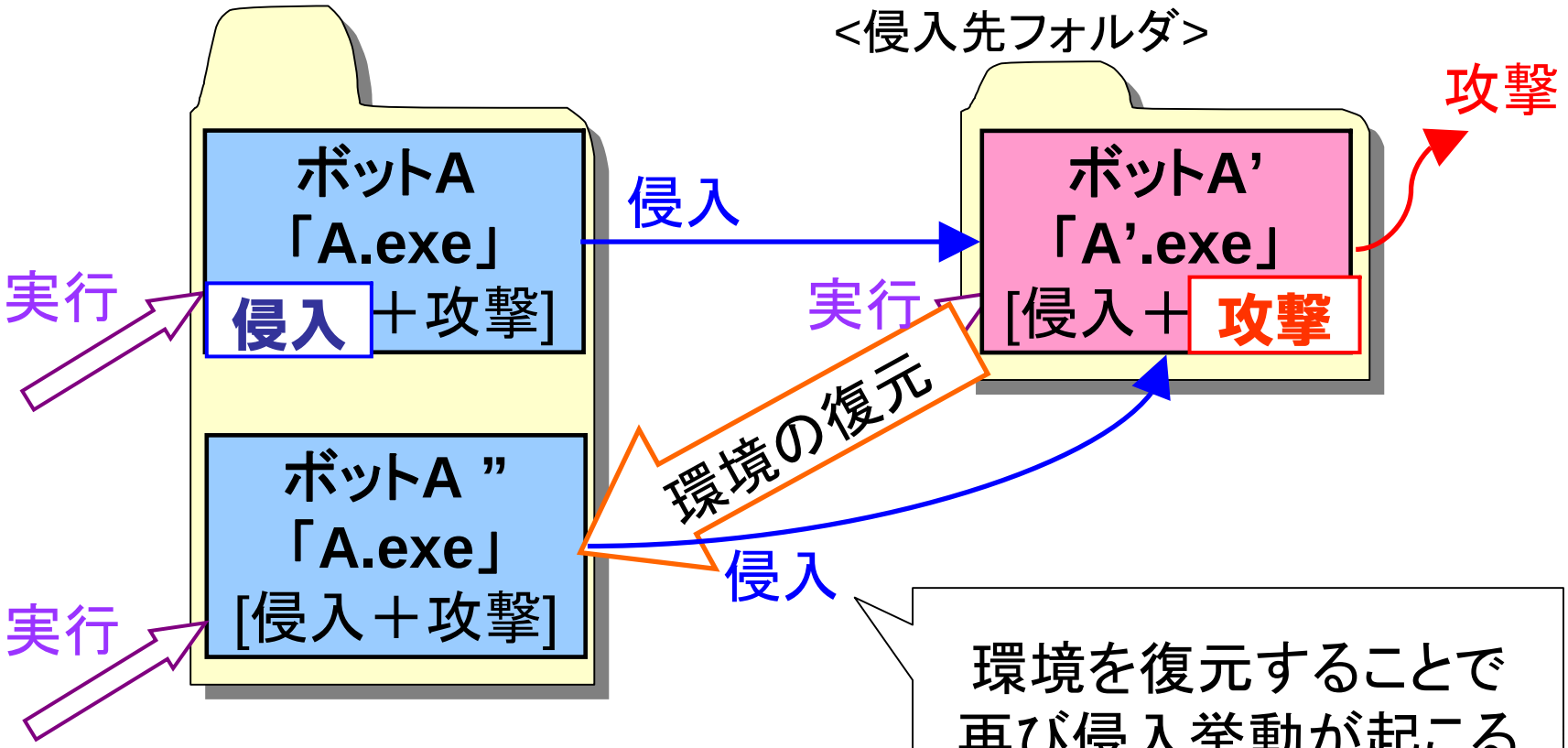
「侵入挙動の反復性によるボット検知方式」

注目している特徴

- ボット(マルウェア)は侵入と攻撃の両機能を併せ持つ
- ボット(マルウェア)は実行環境によって挙動を変化させる

既存方式（狭くチェックする方式）～ボットの機能の使い分け～

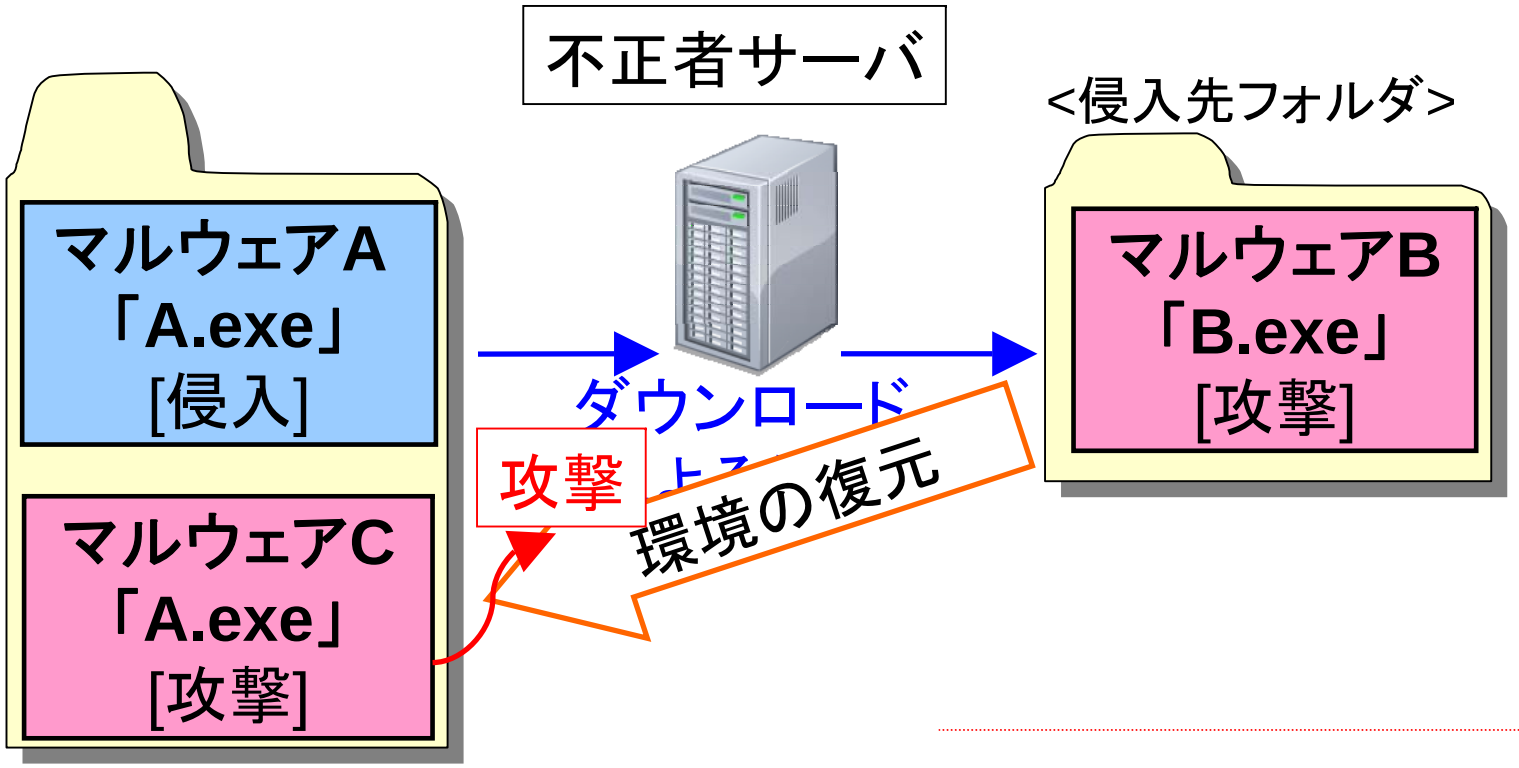
Nish!gaki Laboratory



既存方式（狭くチェックする方式）～検知漏れ～

Nish!gaki Laboratory

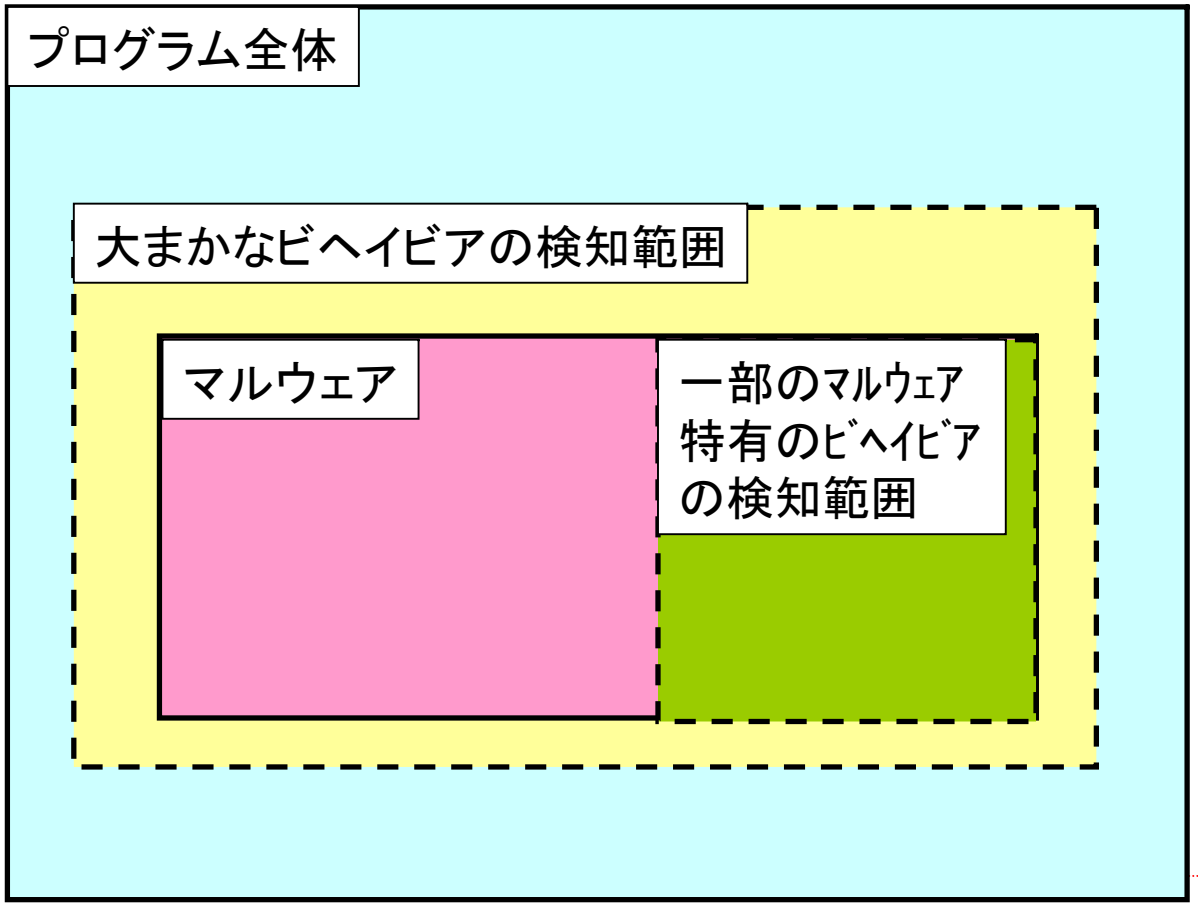
- 侵入と攻撃の機能を別々のファイルに持つ検体を検知することができない.
  - 例. ダウンローダ



ビヘイビアを定義する際の課題

Nish!gaki Laboratory

一部のマルウェア特有のビヘイビアでは，検知から逃れるマルウェアが存在する



誤検知

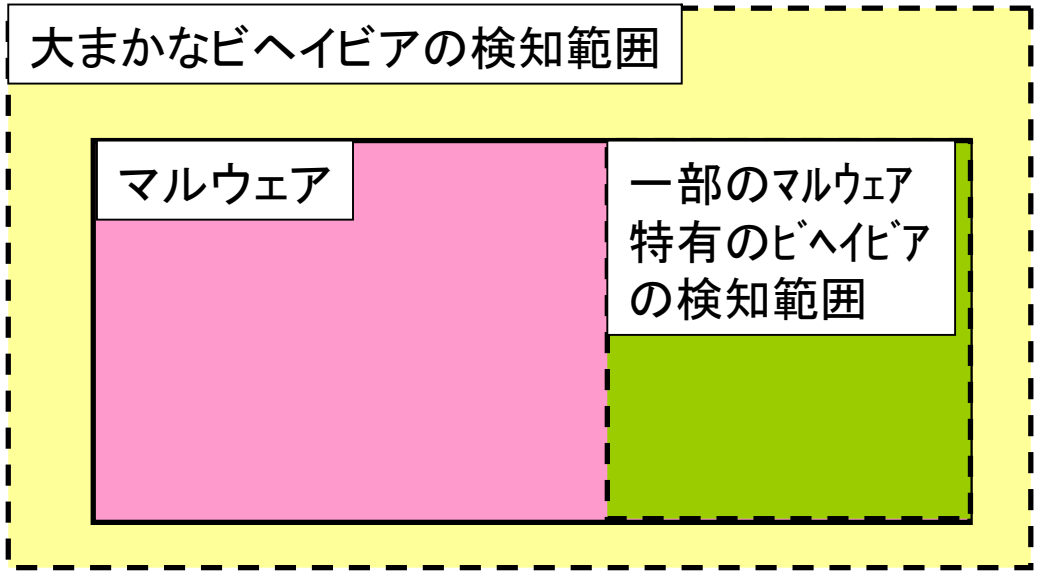
トレードオフ

検知漏れ

特徴を捉えている分
検知対象も絞られる

誤検知なしのビヘイビアブロッキング

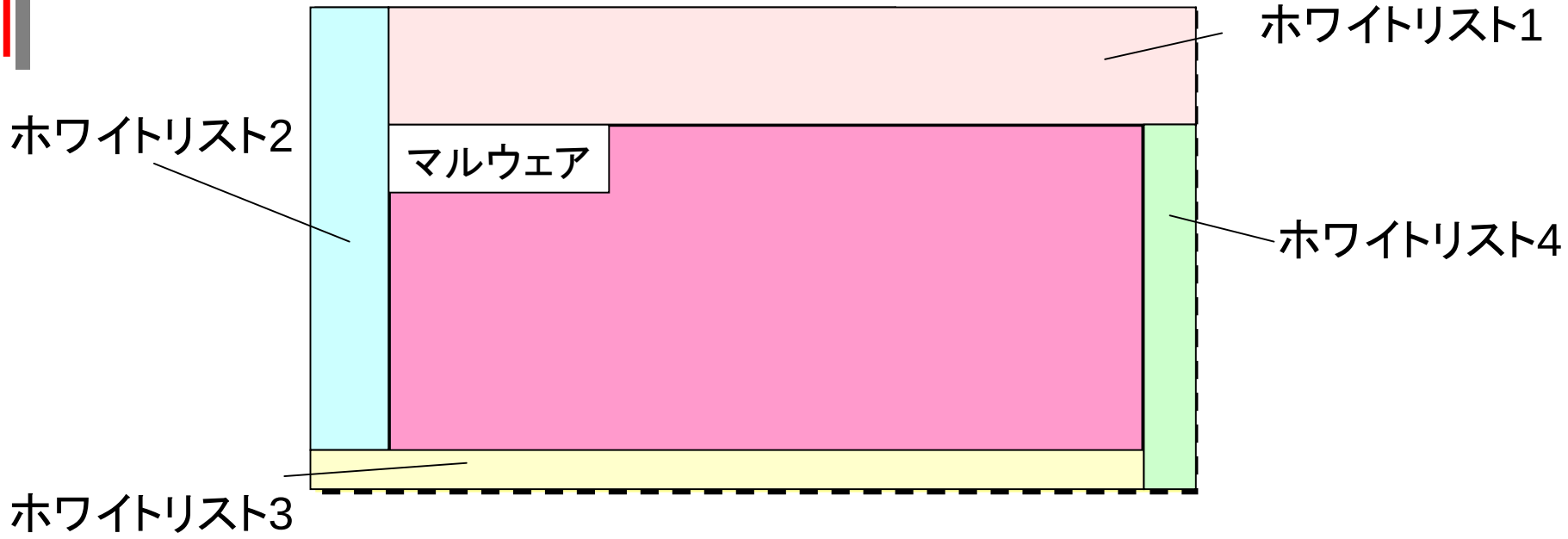
Nish!gaki Laboratory



- 誤検知なしですべてのマルウェアを検知する方法
  1. 大まかなビヘイビアから誤検知を減らしていく
  2. 誤検知のないビヘイビアを組み合わせる

誤検知なしのビヘイビアブロッキング

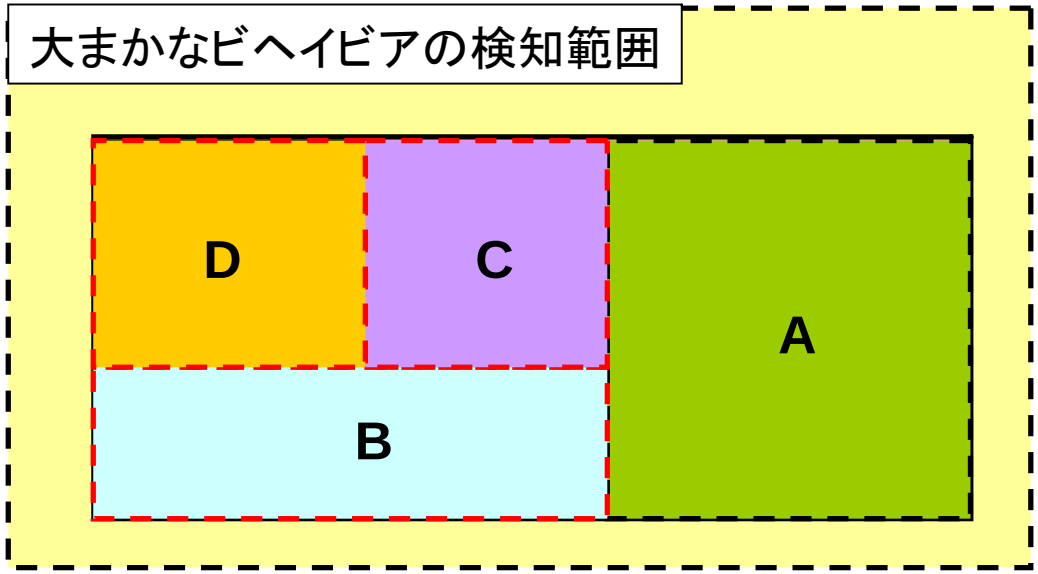
Nish!gaki Laboratory



1. 大まかなビヘイビアから誤検知を減らしていく
   - 正規プログラムの挙動からホワイトリストを作成し，誤検知を減らす

誤検知なしのビヘイビアブロッキング

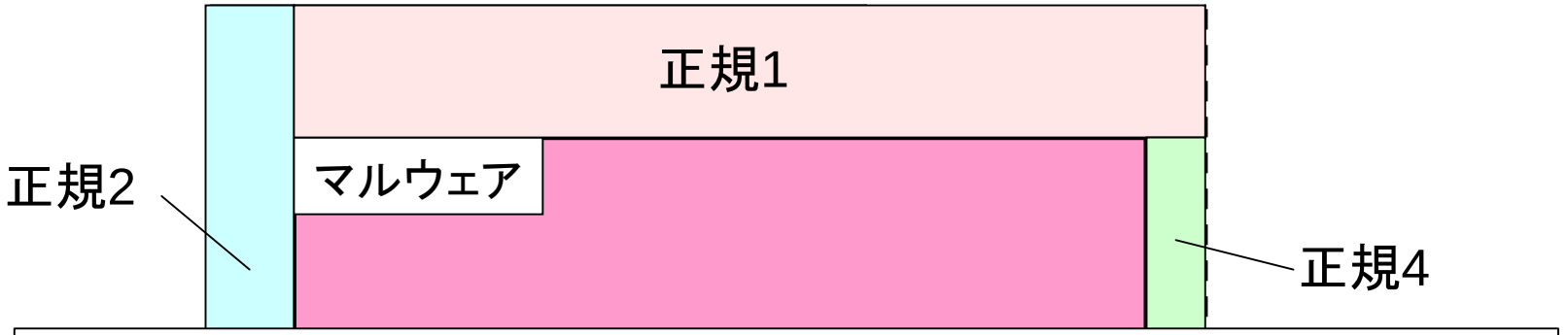
Nish!gaki Laboratory



2. 誤検知のないビヘイビアを組み合わせる
   - 同様に誤検知のない検知方式を検討していくことで，誤検知無くのマルウェアを検知できる

誤検知なしのビヘイビアブロッキング

Nish!gaki Laboratory

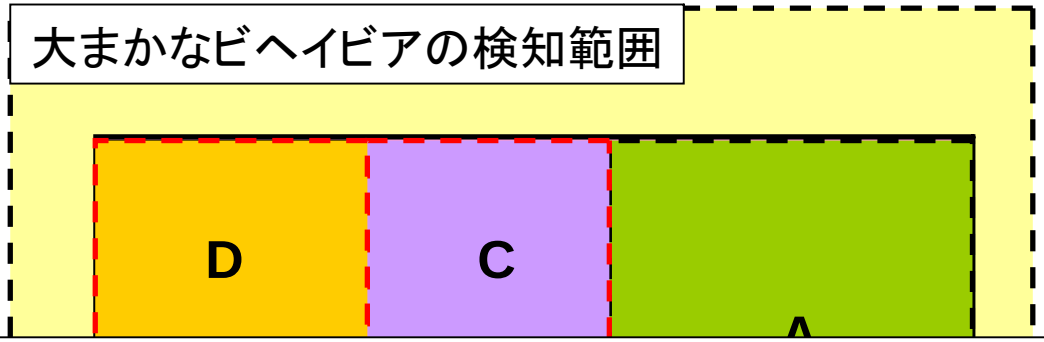


- マルウェアは正規の動作をマネすることも可能
  ⇒容易に回避されてしまう

1. 大まかなビヘイビアから誤検知を減らしていく
   - 正規プログラムの挙動からホワイトリストを作成し，誤検知を減らす

誤検知なしのビヘイビアブロッキング

Nish!gaki Laboratory

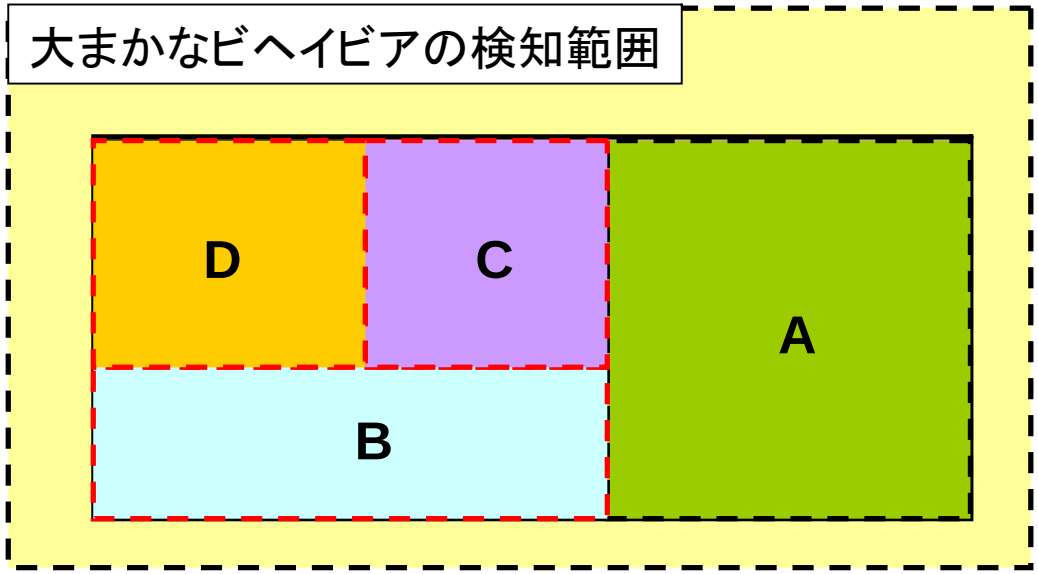


- マルウェア特有の挙動
  ⇒正規プログラムが真似することは無い

2. 誤検知のないビヘイビア[illegible]組み合わせる
   - 同様に誤検知のない検知方式を検討していくことで，誤検知無くのマルウェアを検知できる

誤検知なしのビヘイビアブロッキング

Nish!gaki Laboratory



2. 誤検知のないビヘイビアを組み合わせる

ビヘイビアの分類を行い，各分類ごとに本質を見極め，
検知方式を敷き詰めていく必要がある

分類図の作成

Nish!gaki Laboratory

- 今回はビヘイビアの分類を行い，分類図を作成する
  - 今後検討すべき課題を明らかにする.

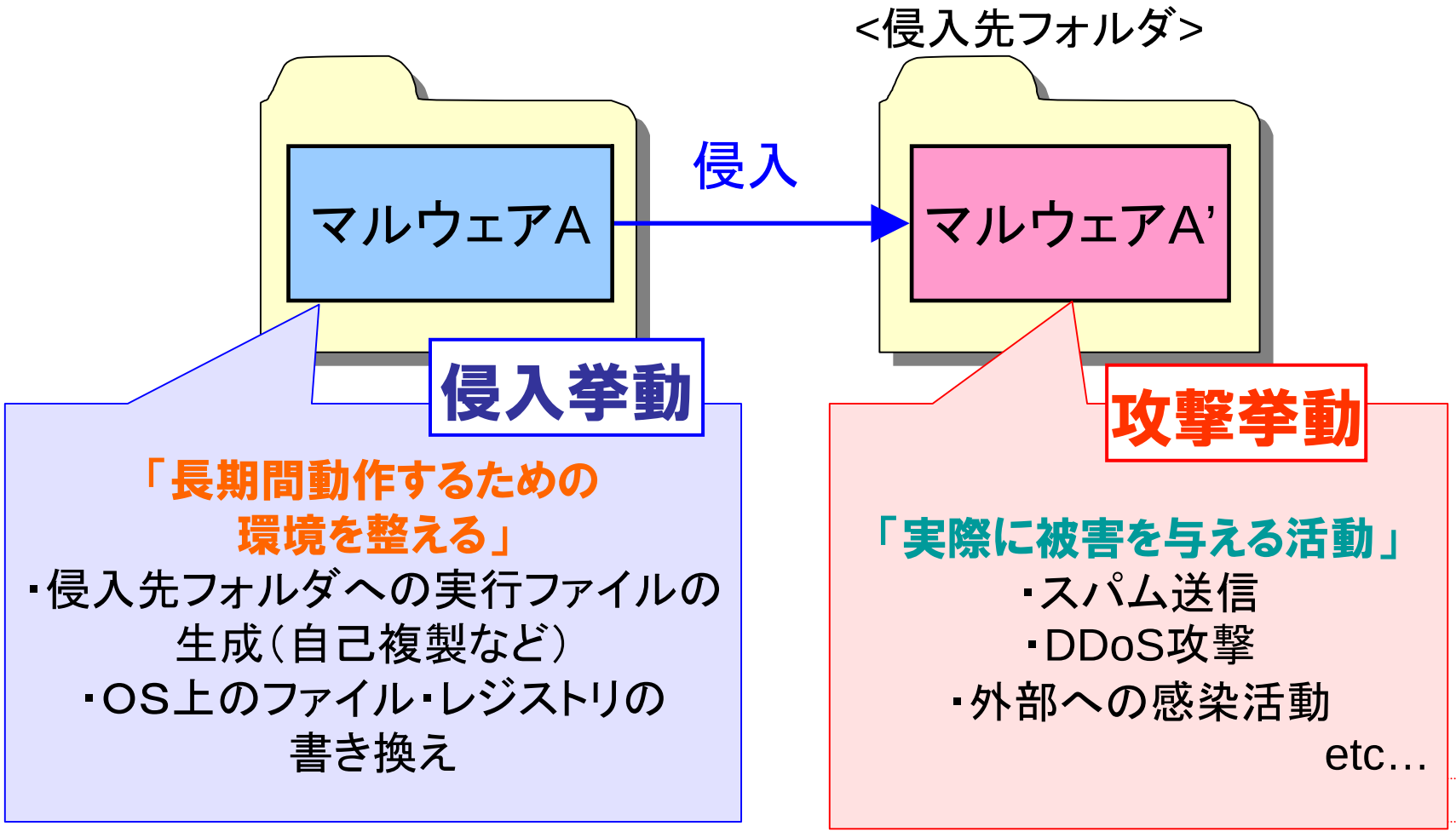
- 闇雲に分類を行う⇒効率が悪い
  何に基づいて分類をしていくべきか？

そこで

- マルウェアの挙動を分析し，どのような観点から分類を行うかを決める.

マルウェアの挙動の分析

Nish!gaki Laboratory



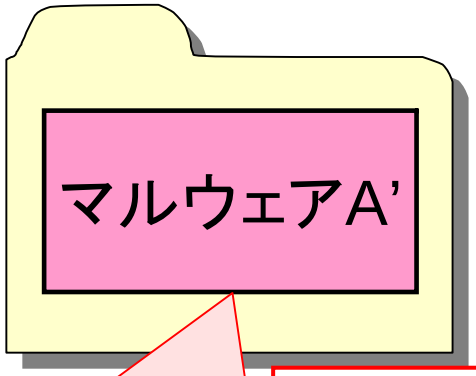
攻撃挙動の特徴

Nish!gaki Laboratory

攻撃挙動

・マルウェアの目的によって、
攻撃方法が異なる
・マルウェア製作者の
任意のタイミングで行われ
るため、**観測が困難**

<侵入先フォルダ>



攻撃挙動

「実際に被害を与える活動」
・スパム送信
・DDoS攻撃
・外部への感染活動
etc…

侵入挙動の特徴

Nish!gaki Laboratory

侵入挙動(＝自動実行登録)に注目する

侵入挙動

「長期間動作するための環境を整える」

・侵入先フォルダへの実行ファイルの生成(自己複製など)

・OS上のファイル・レジストリの書き換え

侵入挙動

・常駐するためには、**自動実行登録が必須**

・初期実行直後に行う必要があり、**観測が容易**

今回使用するビヘイビア

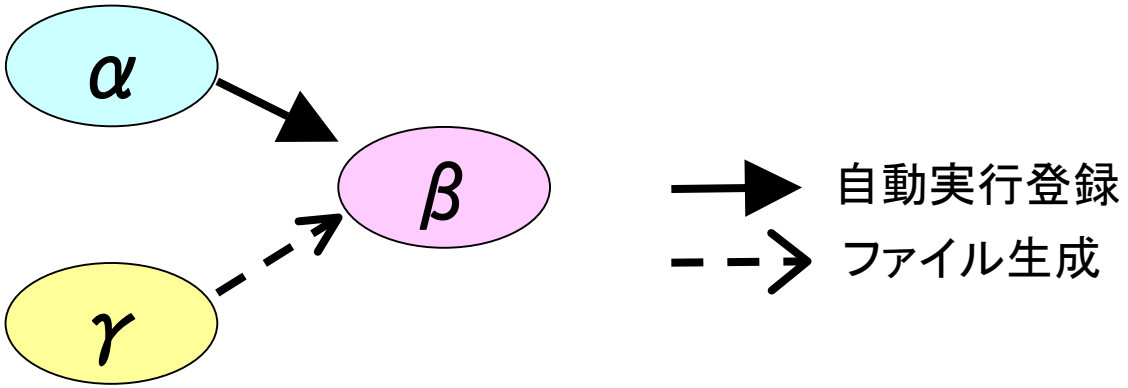
Nish!gaki Laboratory

1. 検知対象の狭いビヘイビア
2. 侵入挙動に関するビヘイビア

- 侵入挙動の反復性によるボット検知方式
- 連携感染型マルウェアの検知

連携感染型マルウェア

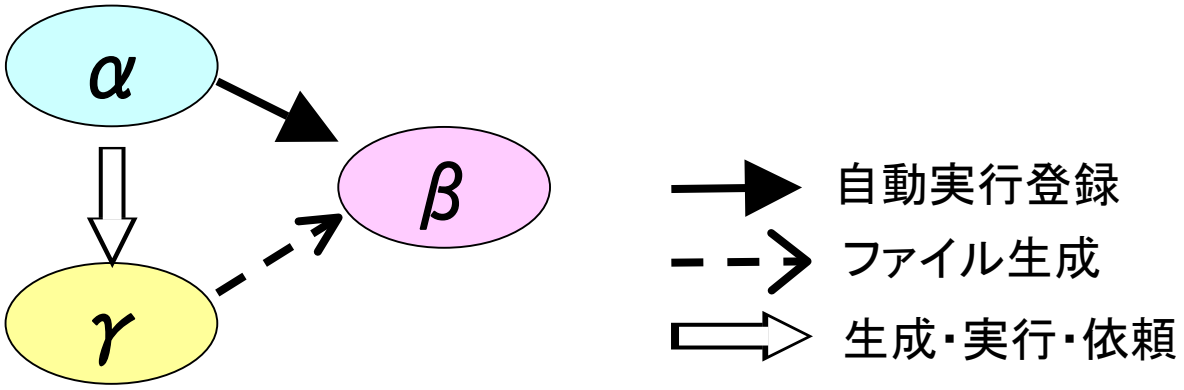
- 複数の脆弱性を利用してPCに感染(連携感染)



- 正規プログラムの場合はこのように手間をかける必要は無い.

連携感染型マルウェア

- 正規プログラムで似たようなことが起こる場合



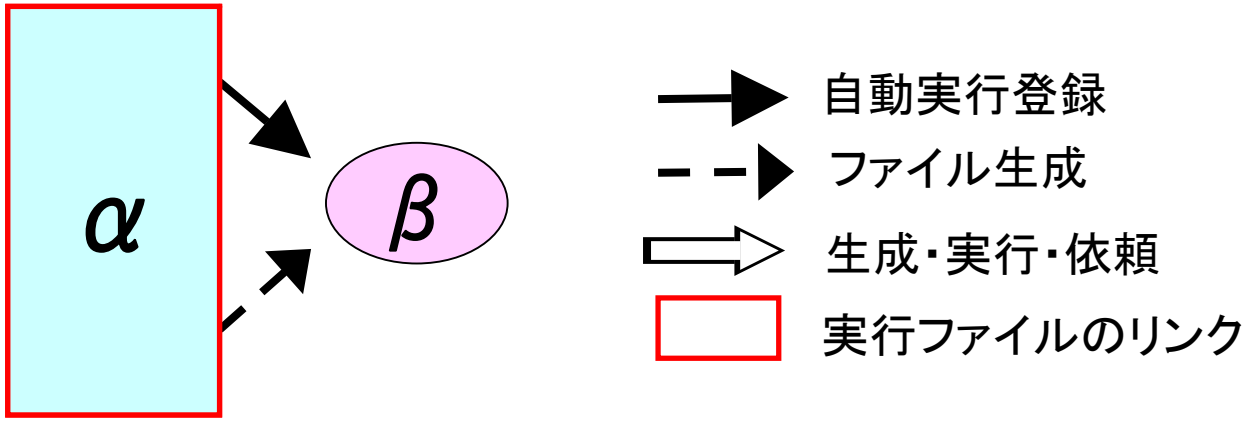
実際にはいずれか1つのプロセスが
全体を制御している

実行ファイルのリンク

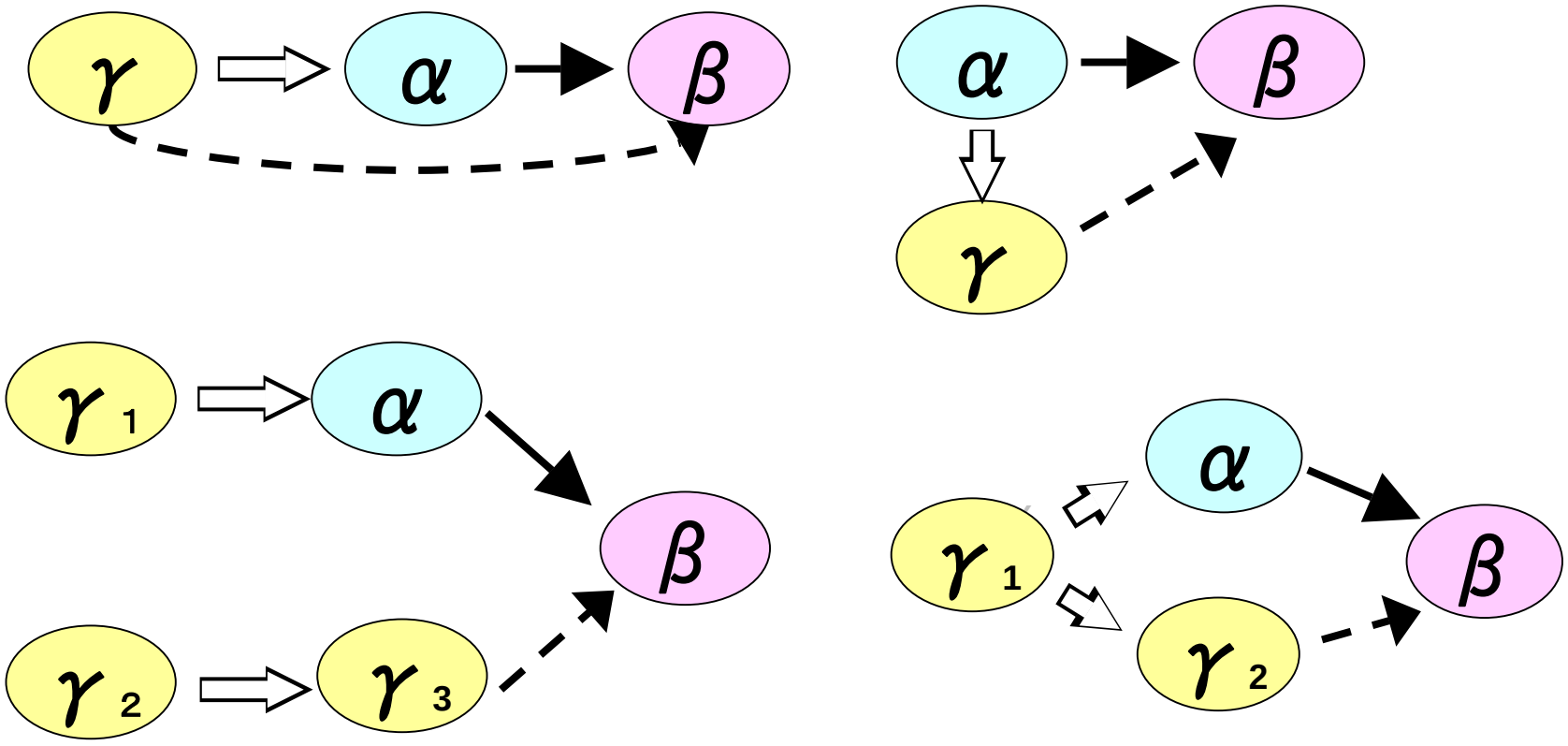
Nish!gaki Laboratory

⇨ 生成・実行・依頼

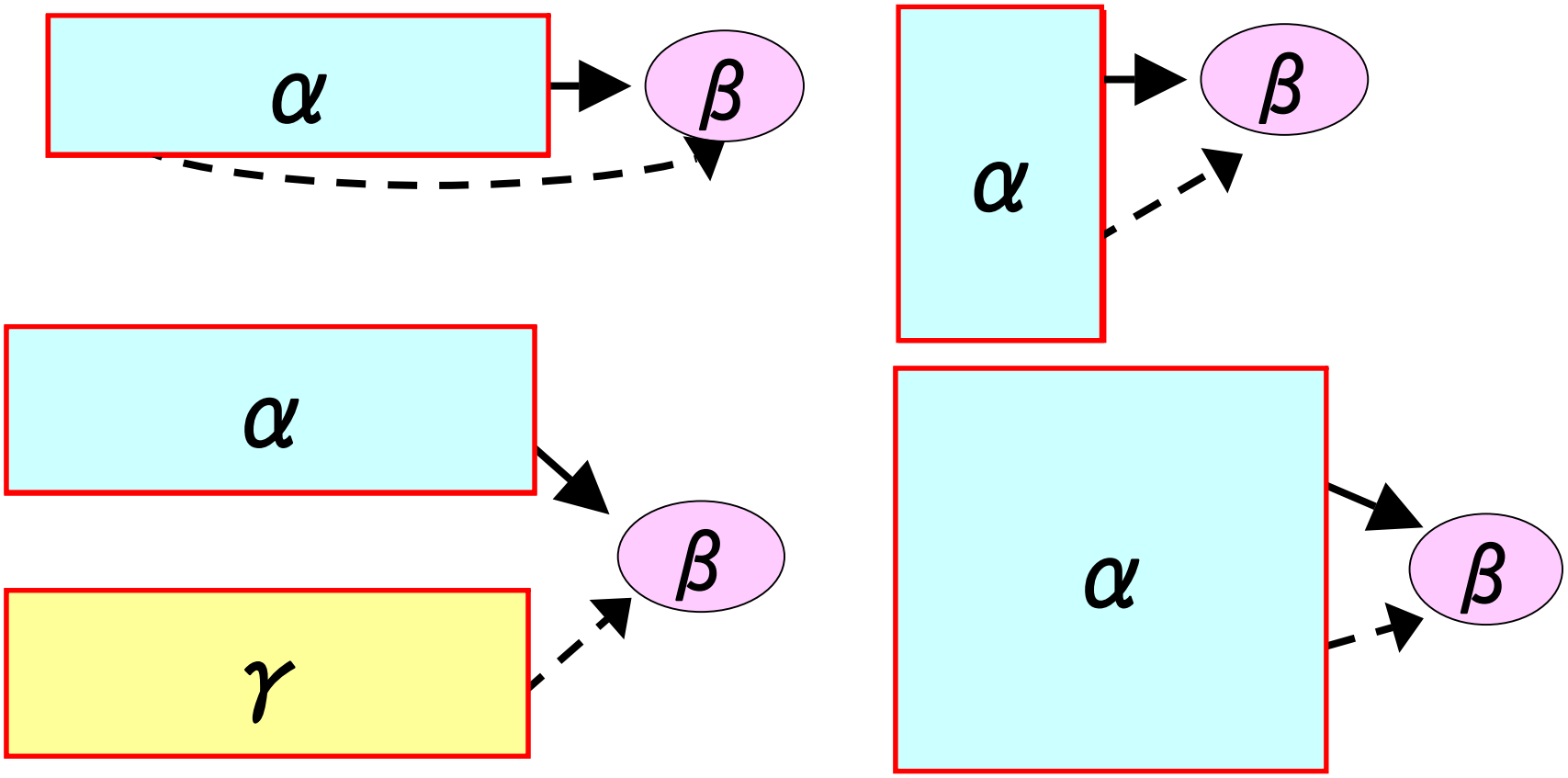
の関係にあるファイル群を1つのグループとみなす



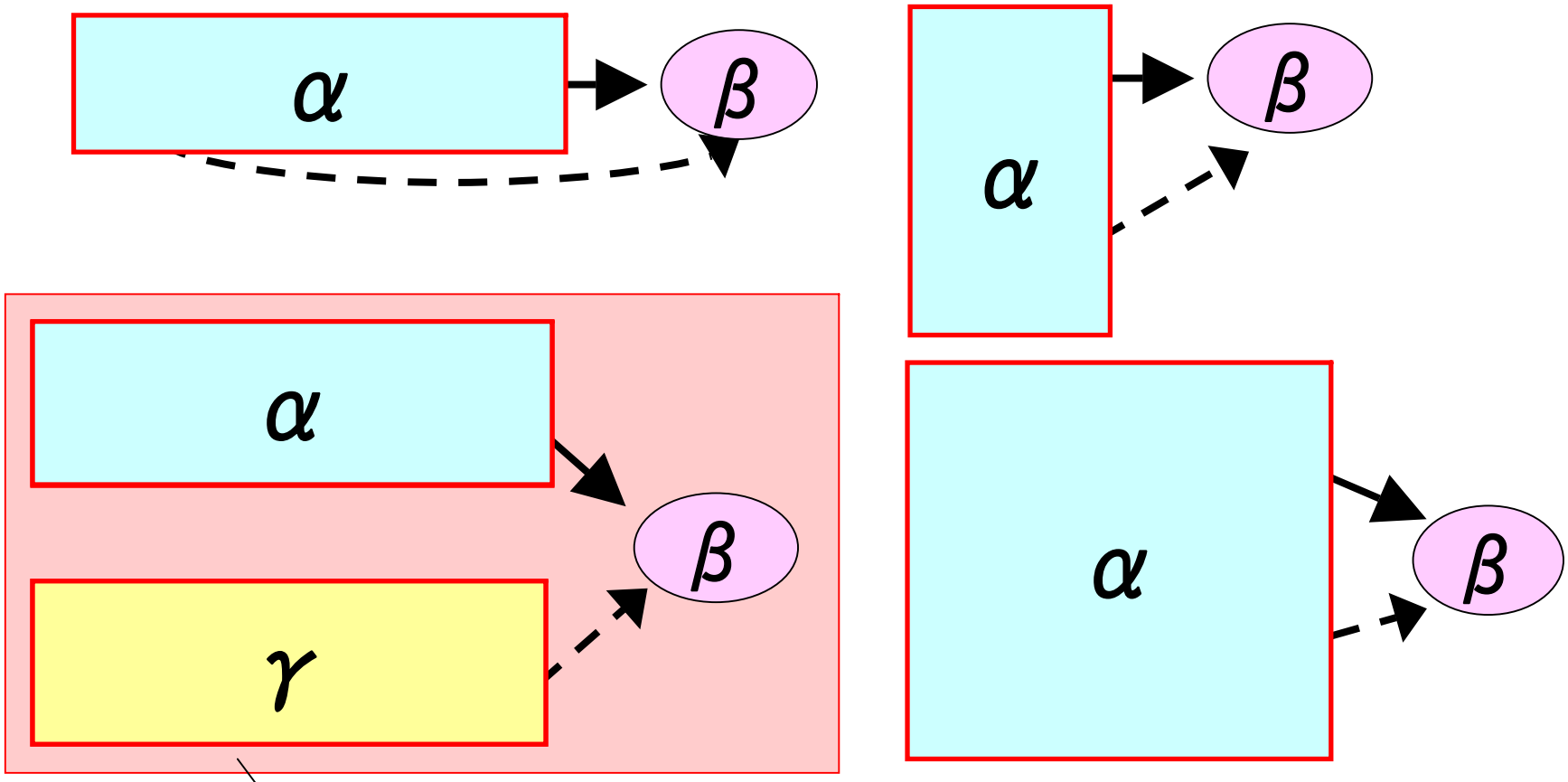
実行のリンクを考慮した分類



実行のリンクを考慮した分類



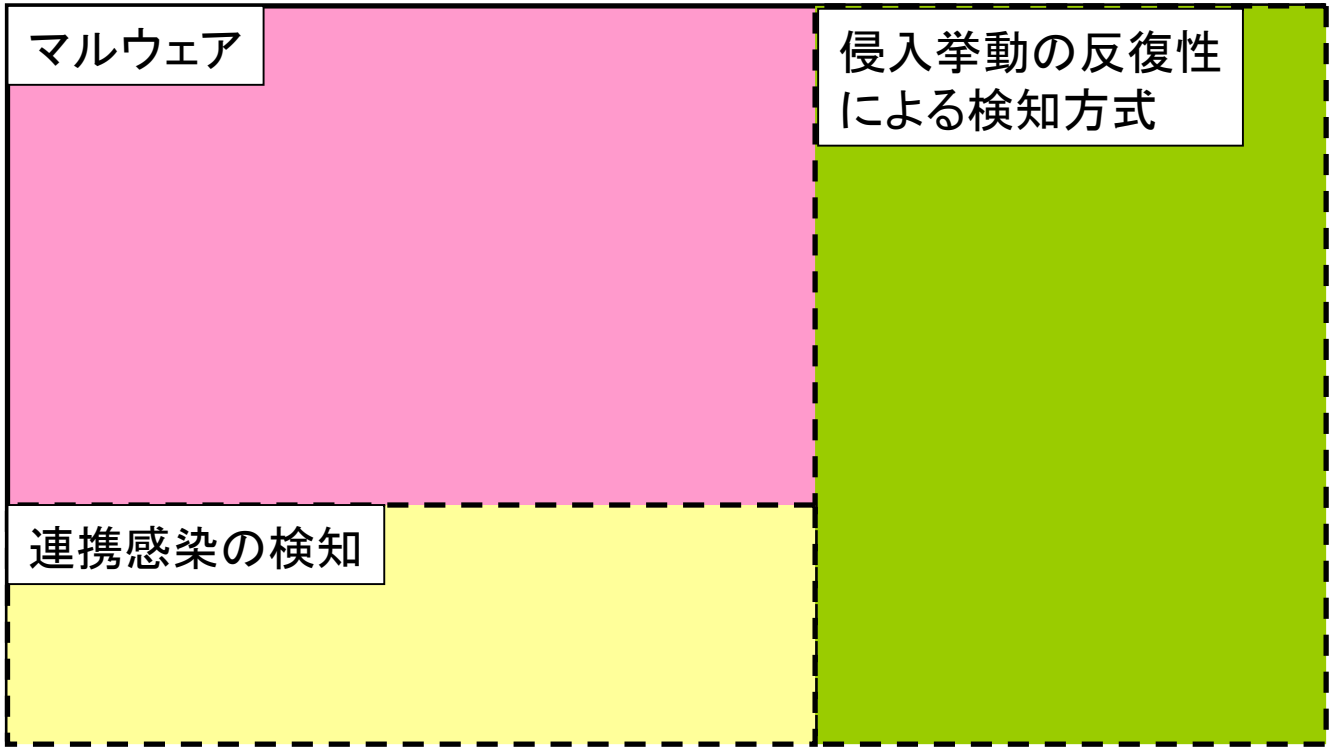
実行のリンクを考慮した分類



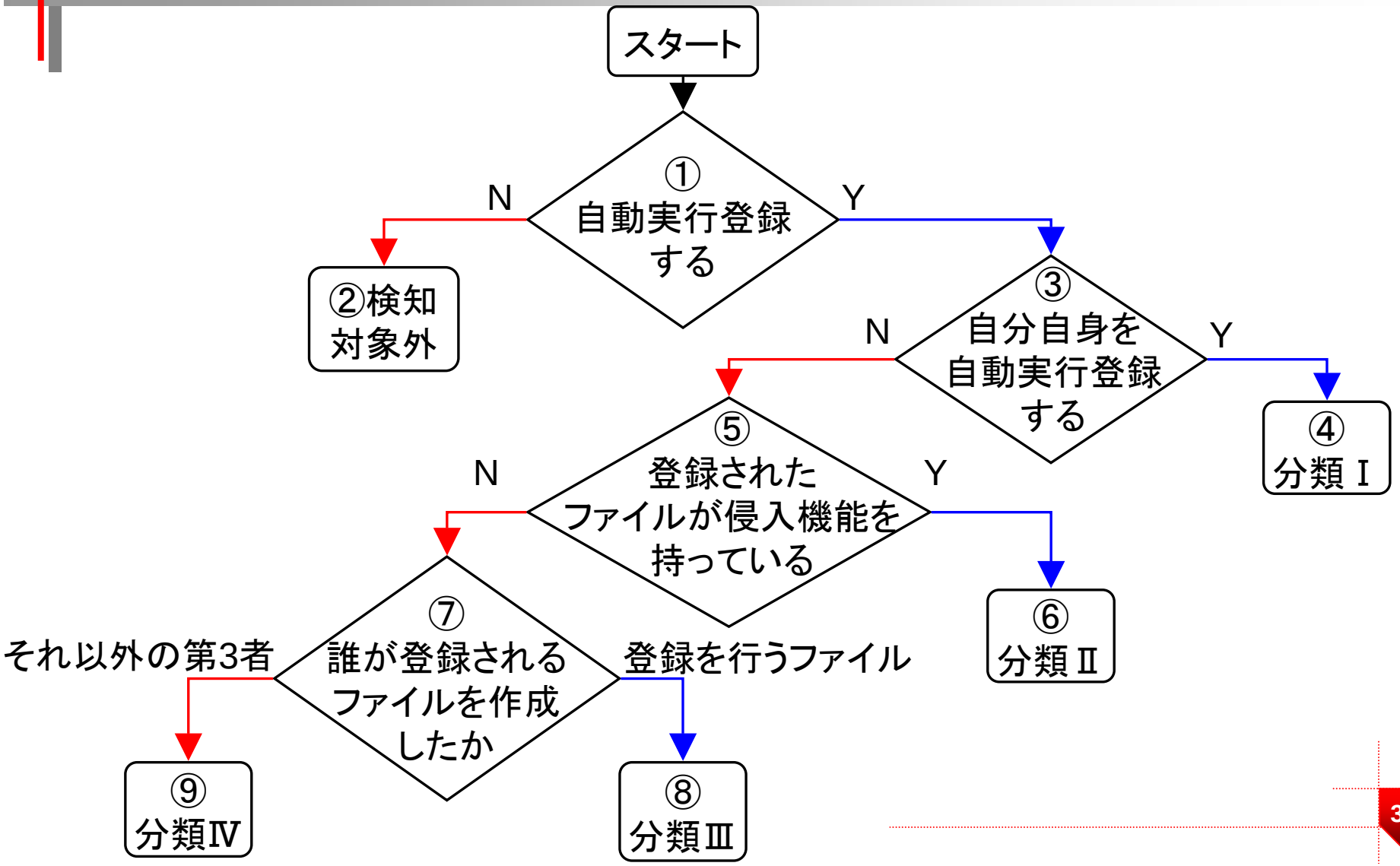
複数の無関係なプロセスが連携している
⇒連携感染

侵入挙動に関連した ビヘイビアブロッキング

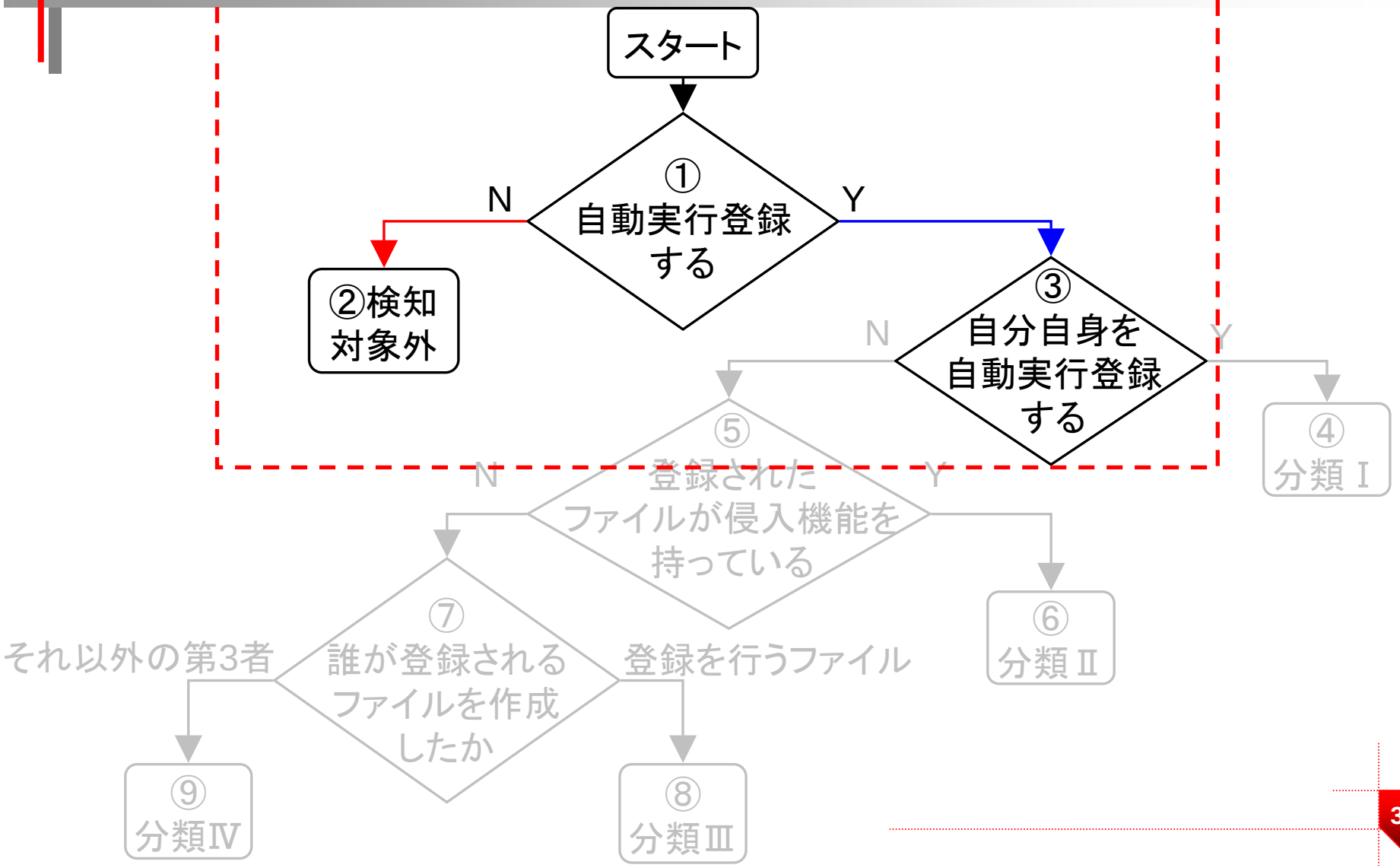
Nish!gaki Laboratory



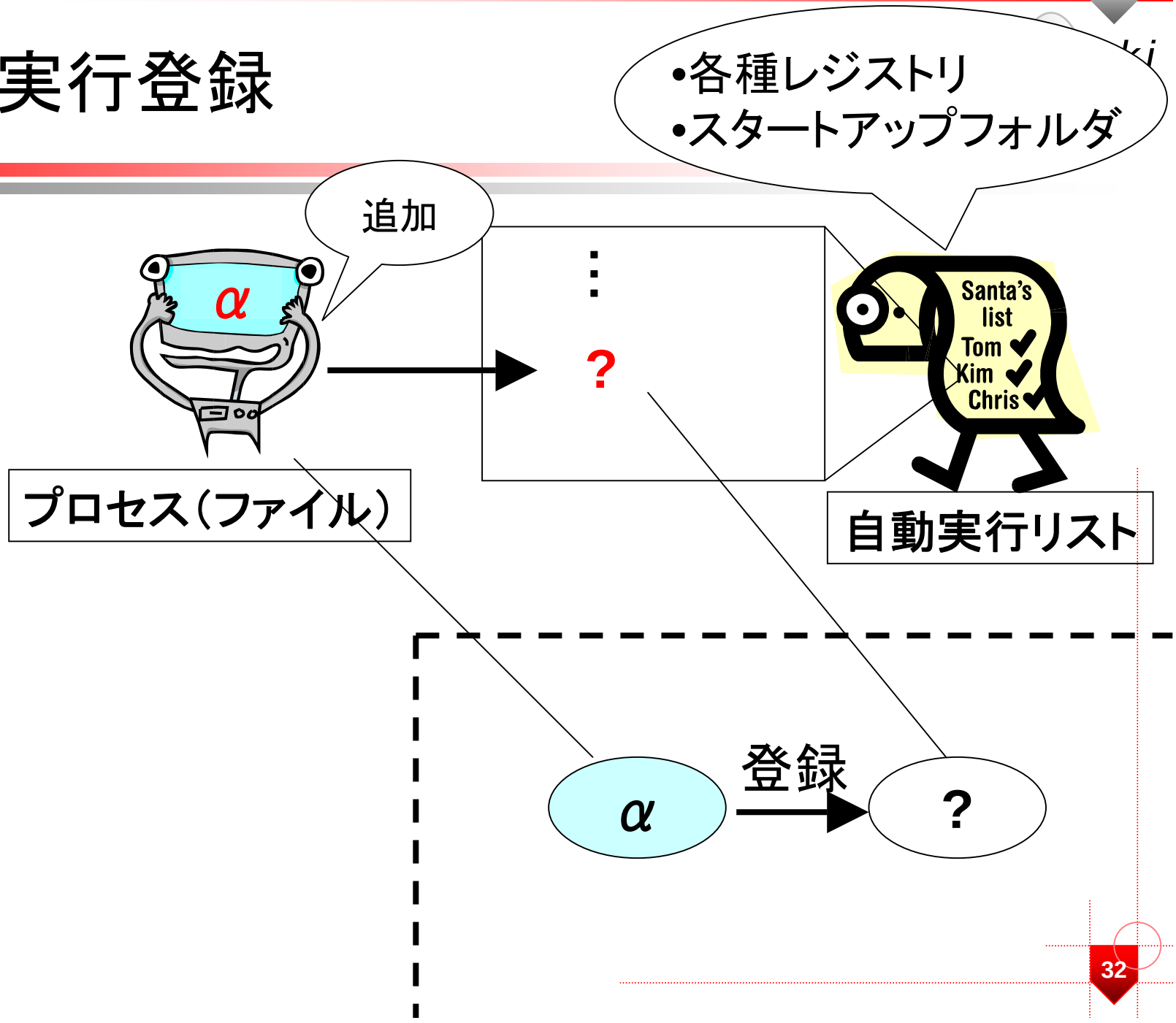
作成した分類図


スタート
①
自動実行登録する
N
Y
②検知対象外
③
自分自身を自動実行登録する
N
Y
④
分類Ⅰ
⑤
登録されたファイルが侵入機能を持っている
N
Y
⑥
分類Ⅱ
⑦
誰が登録されるファイルを作成したか
それ以外の第3者
登録を行うファイル
⑨
分類Ⅳ
⑧
分類Ⅲ

分類手順1

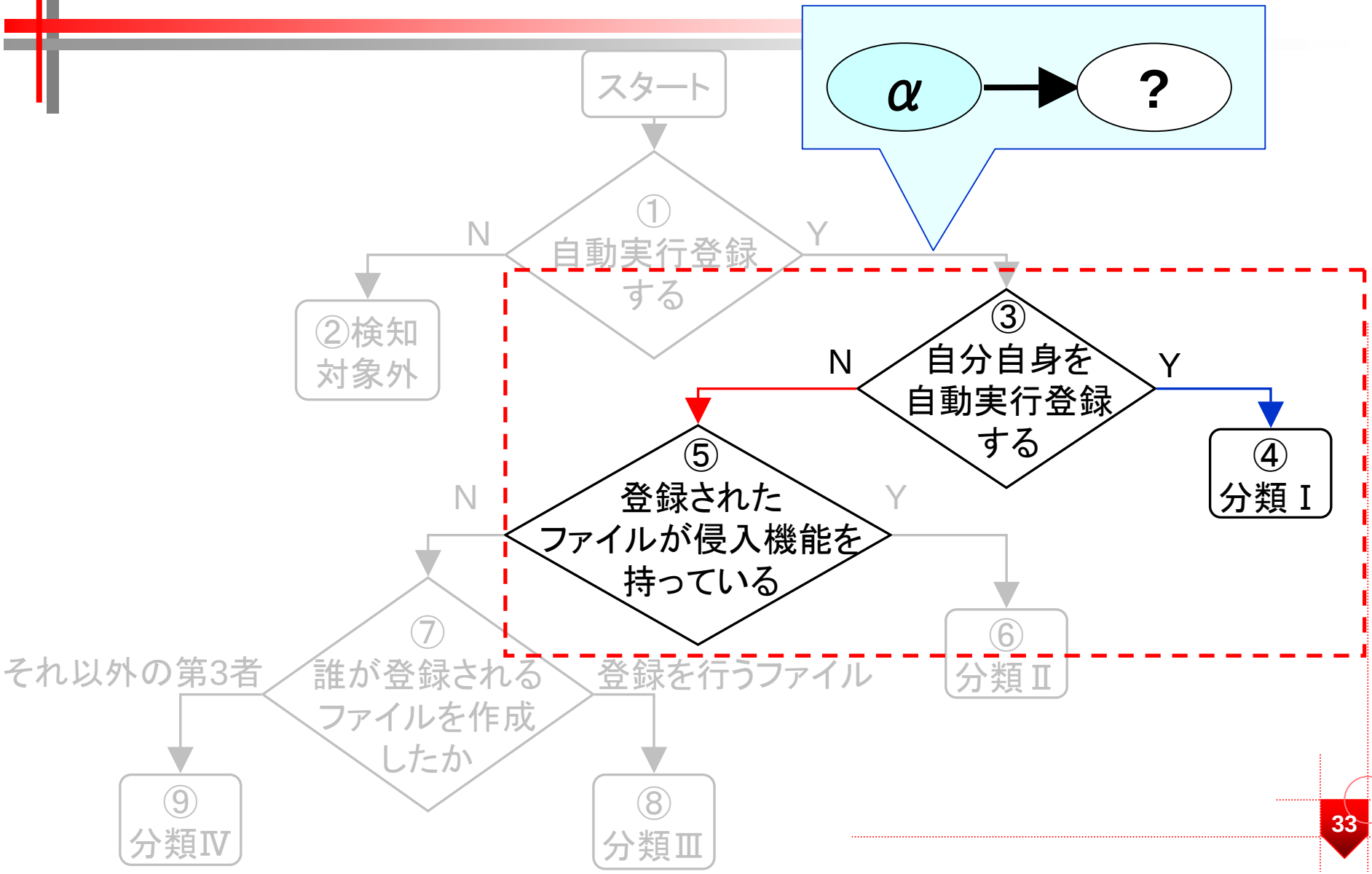

スタート
①
自動実行登録する
N
Y
②検知対象外
③
自分自身を自動実行登録する
N
Y
④
分類Ⅰ
⑤
登録されたファイルが侵入機能を持っている
N
Y
⑥
分類Ⅱ
⑦
誰が登録されるファイルを作成したか
それ以外の第3者
登録を行うファイル
⑧
分類Ⅲ
⑨
分類Ⅳ

自動実行登録


•各種レジストリ
•スタートアップフォルダ
追加
α
?
Santa's list
Tom
Kim
Chris
プロセス(ファイル)
自動実行リスト
α
登録
?

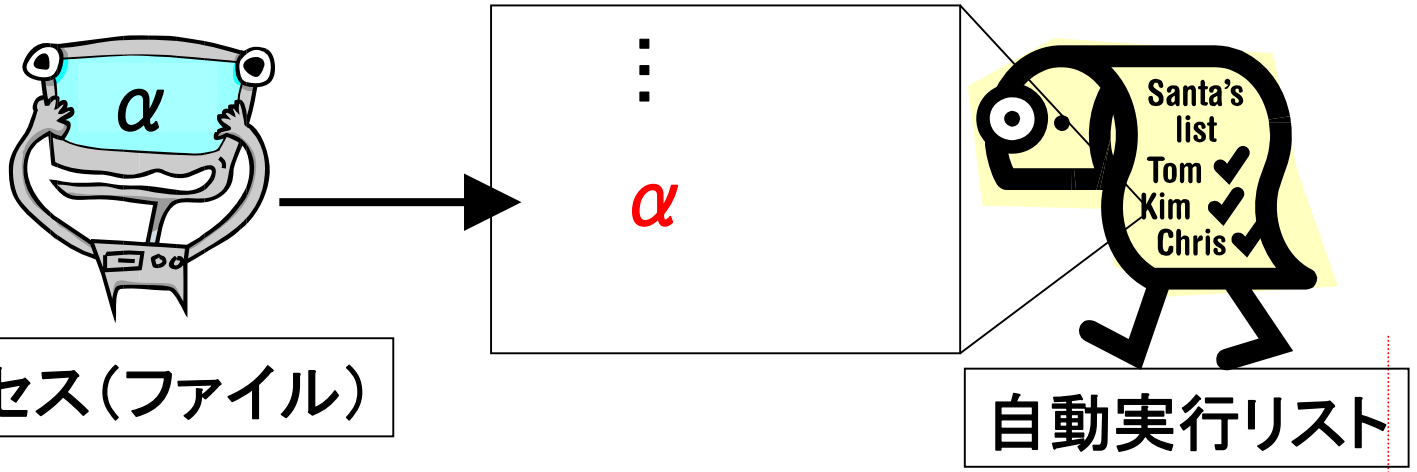
分類手順2

Nish!gaki Laboratory

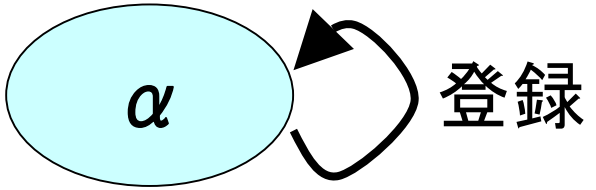


③の分類(④分類Ⅰ)

Nish!gaki Laboratory

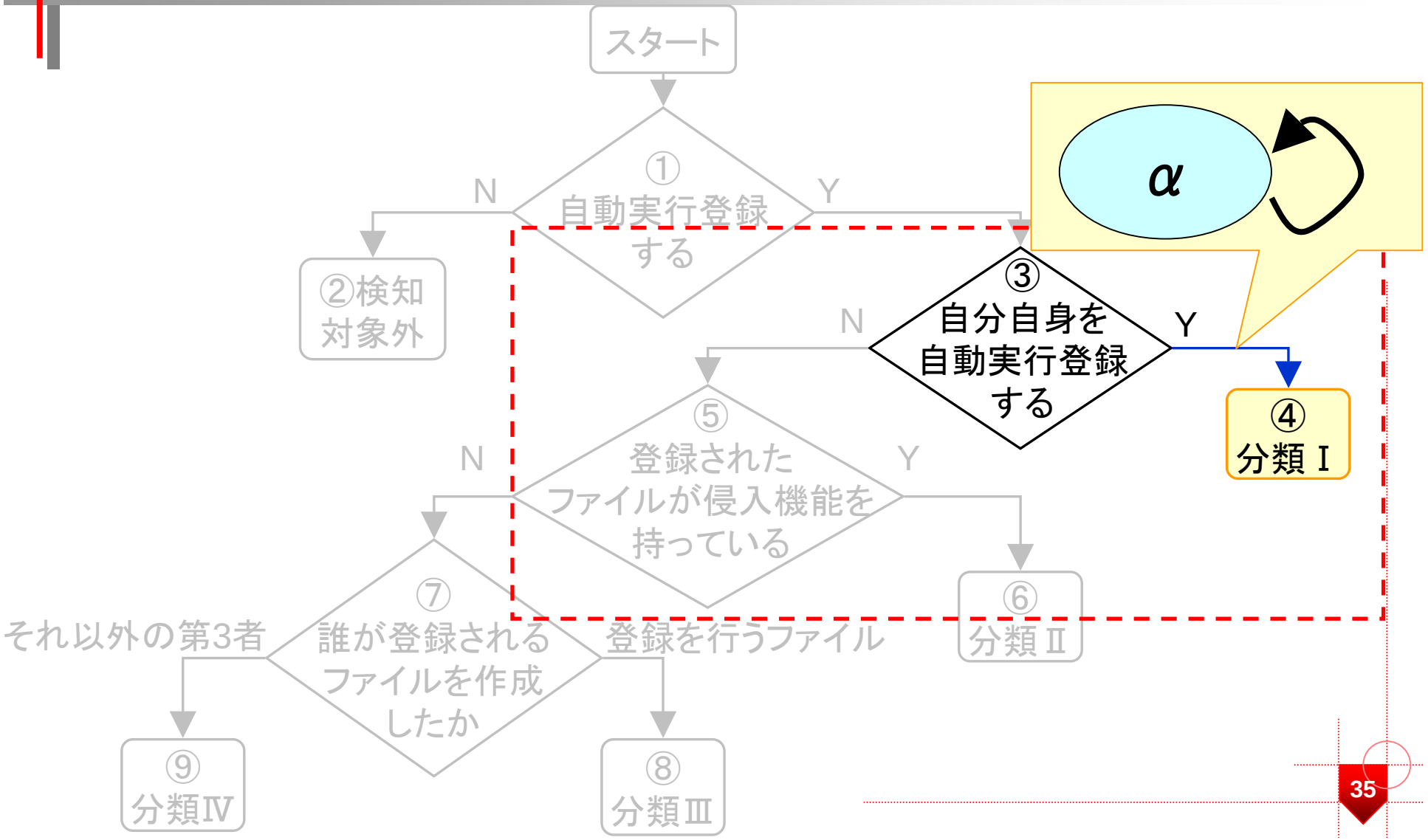


侵入挙動の反復性が見られる

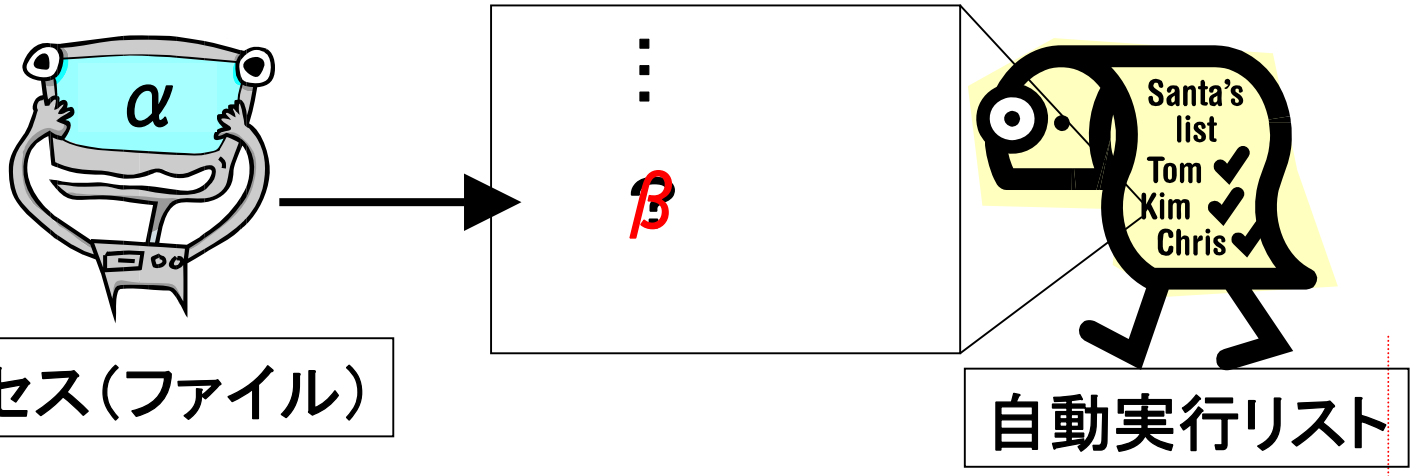


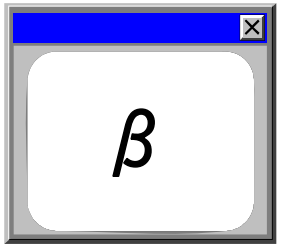
分類手順2

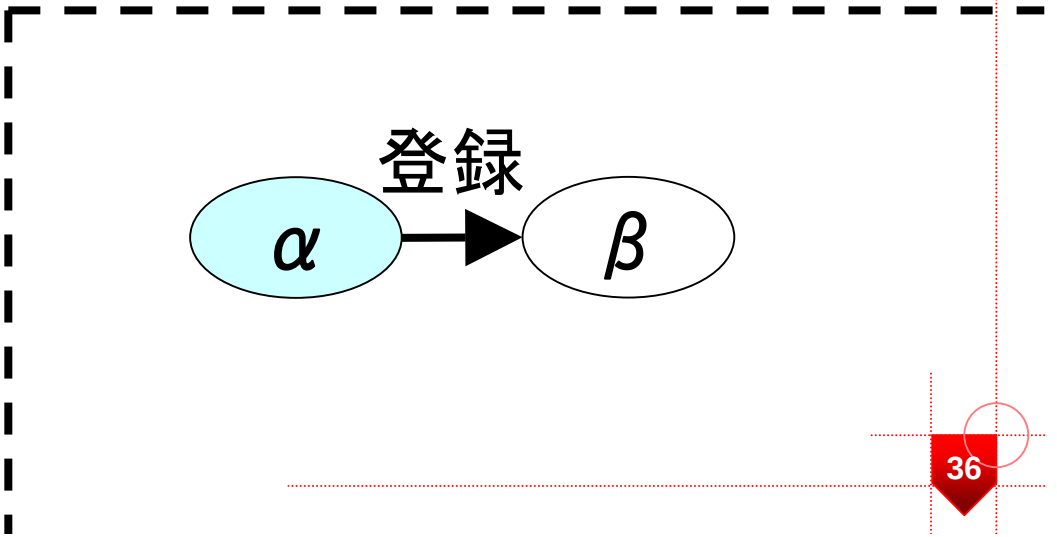
Nish!gaki Laboratory



③の分類(⑤)

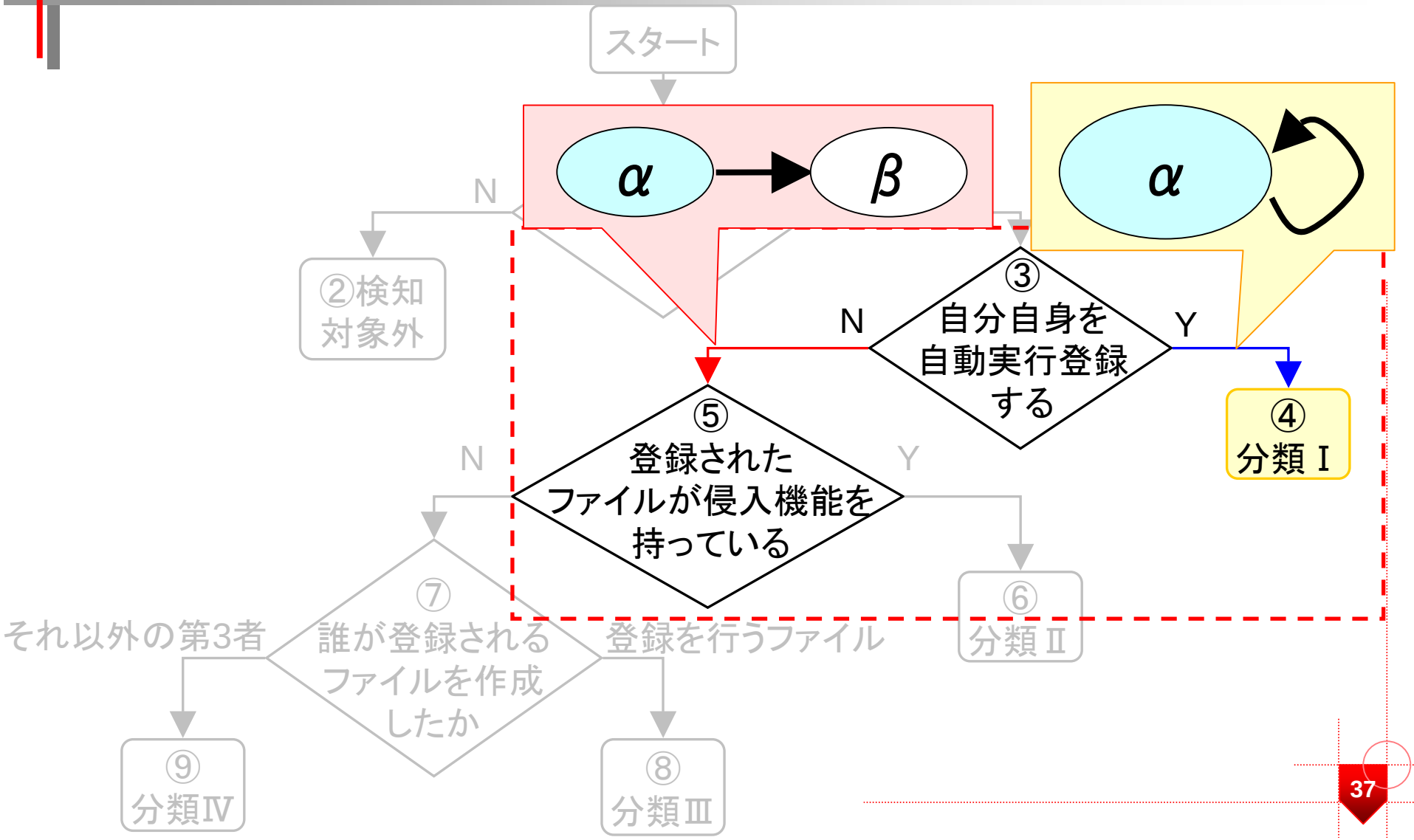




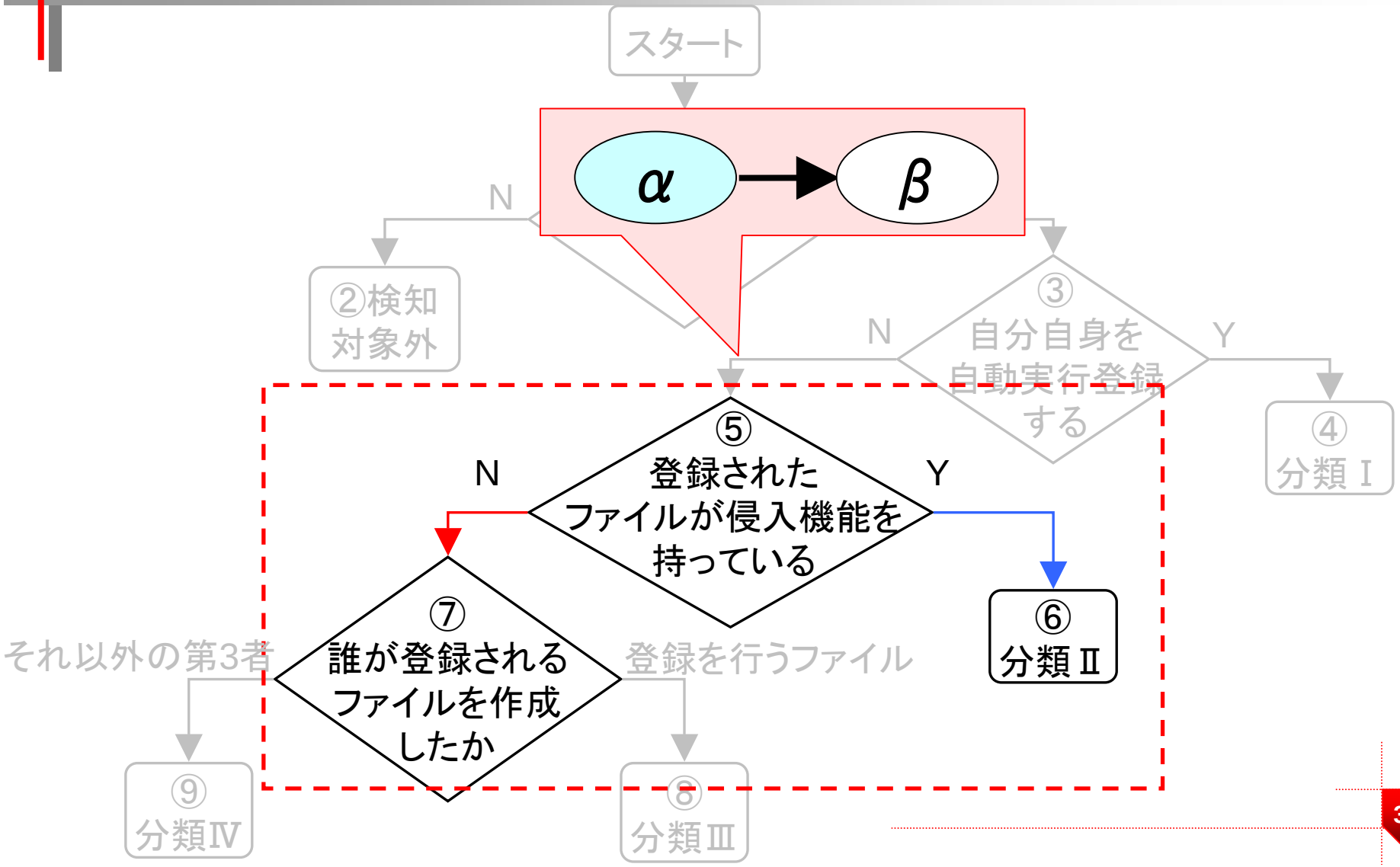


分類手順2

Nish!gaki Laboratory

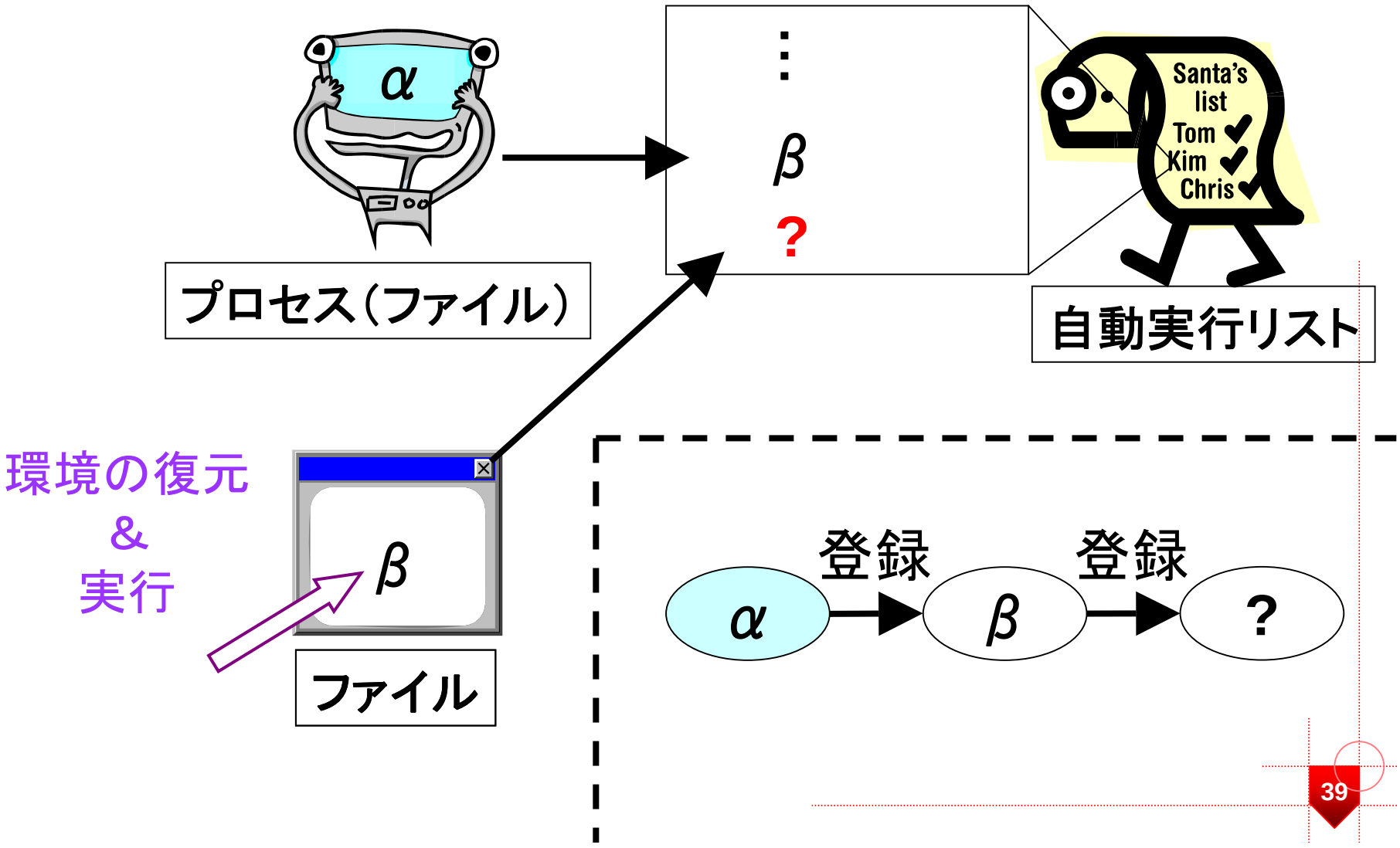


分類手順3

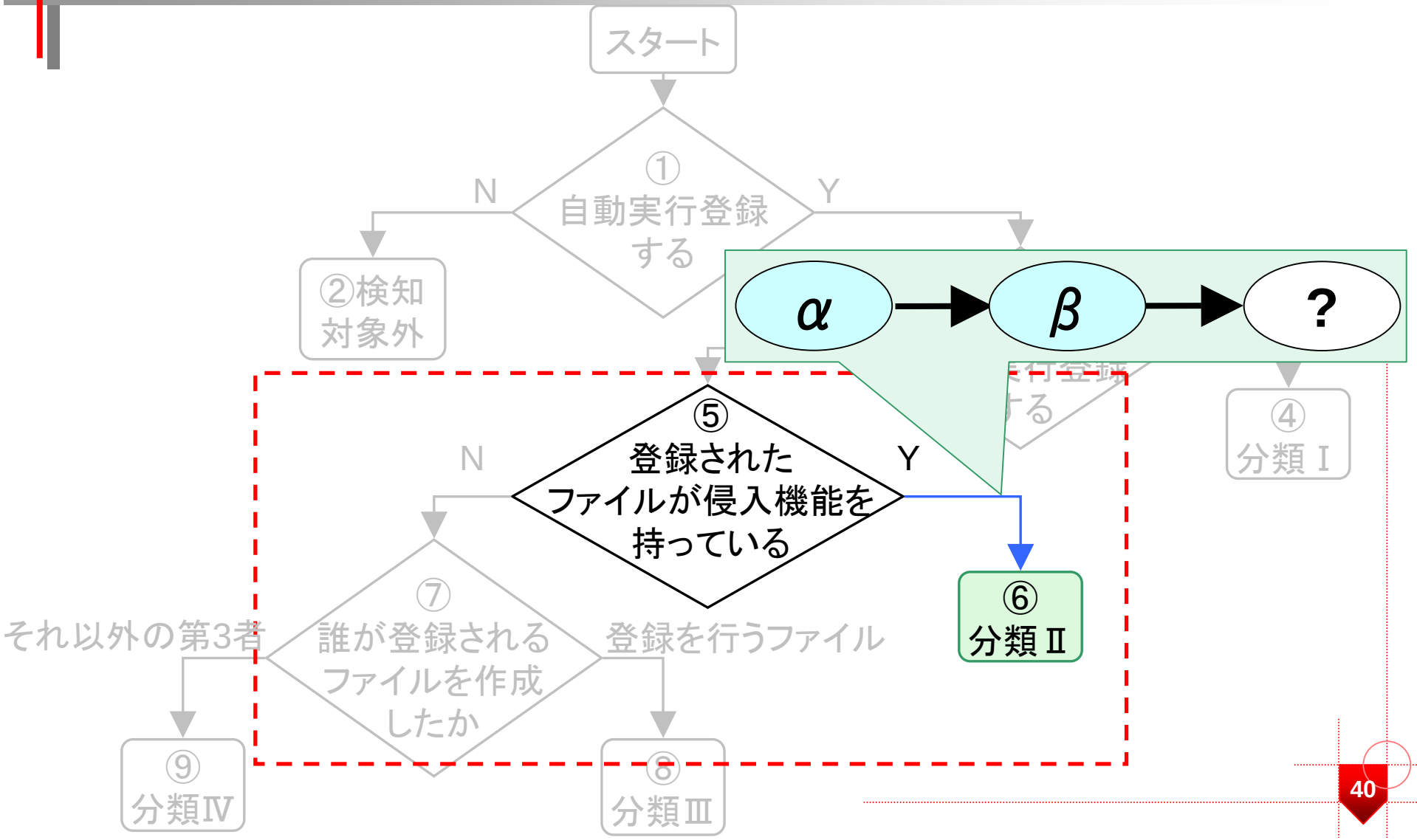

スタート
α
β
N
②検知
対象外
③
自分自身を
自動実行登録
する
N
Y
④
分類Ⅰ
⑤
登録された
ファイルが侵入機能を
持っている
N
Y
⑥
分類Ⅱ
⑦
誰が登録される
ファイルを作成
したか
それ以外の第3者
登録を行うファイル
⑨
分類Ⅳ
⑧
分類Ⅲ

⑤の分類(⑥分類Ⅱ)

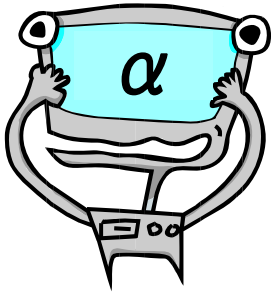
Nish!gaki Laboratory



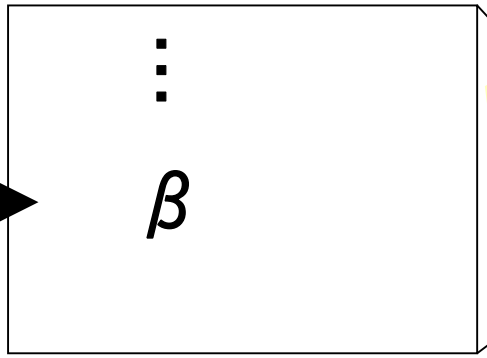
分類手順3


スタート
①
自動実行登録する
N
Y
②検知対象外
α
β
?
④
分類Ⅰ
⑤
登録されたファイルが侵入機能を持っている
N
Y
⑥
分類Ⅱ
⑦
誰が登録されるファイルを作成したか
それ以外の第3者
登録を行うファイル
⑨
分類Ⅳ
⑧
分類Ⅲ

⑤の分類(⑦)



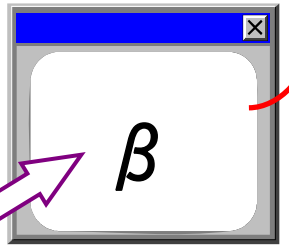
プロセス(ファイル)



自動実行リスト

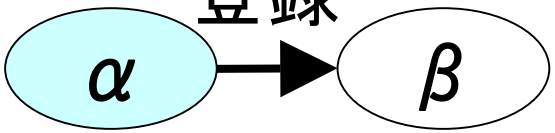
攻撃

環境の復元
&
実行

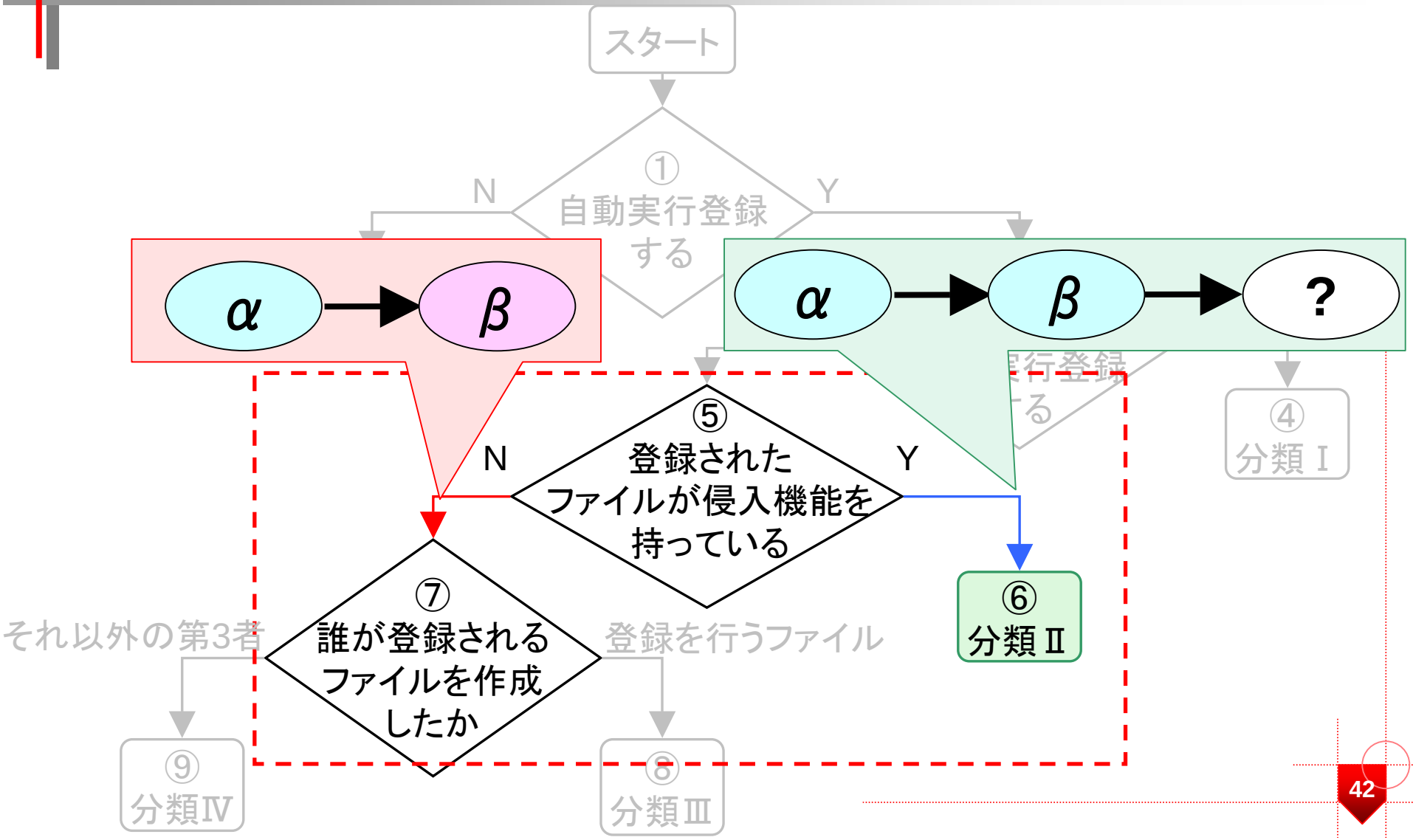


ファイル

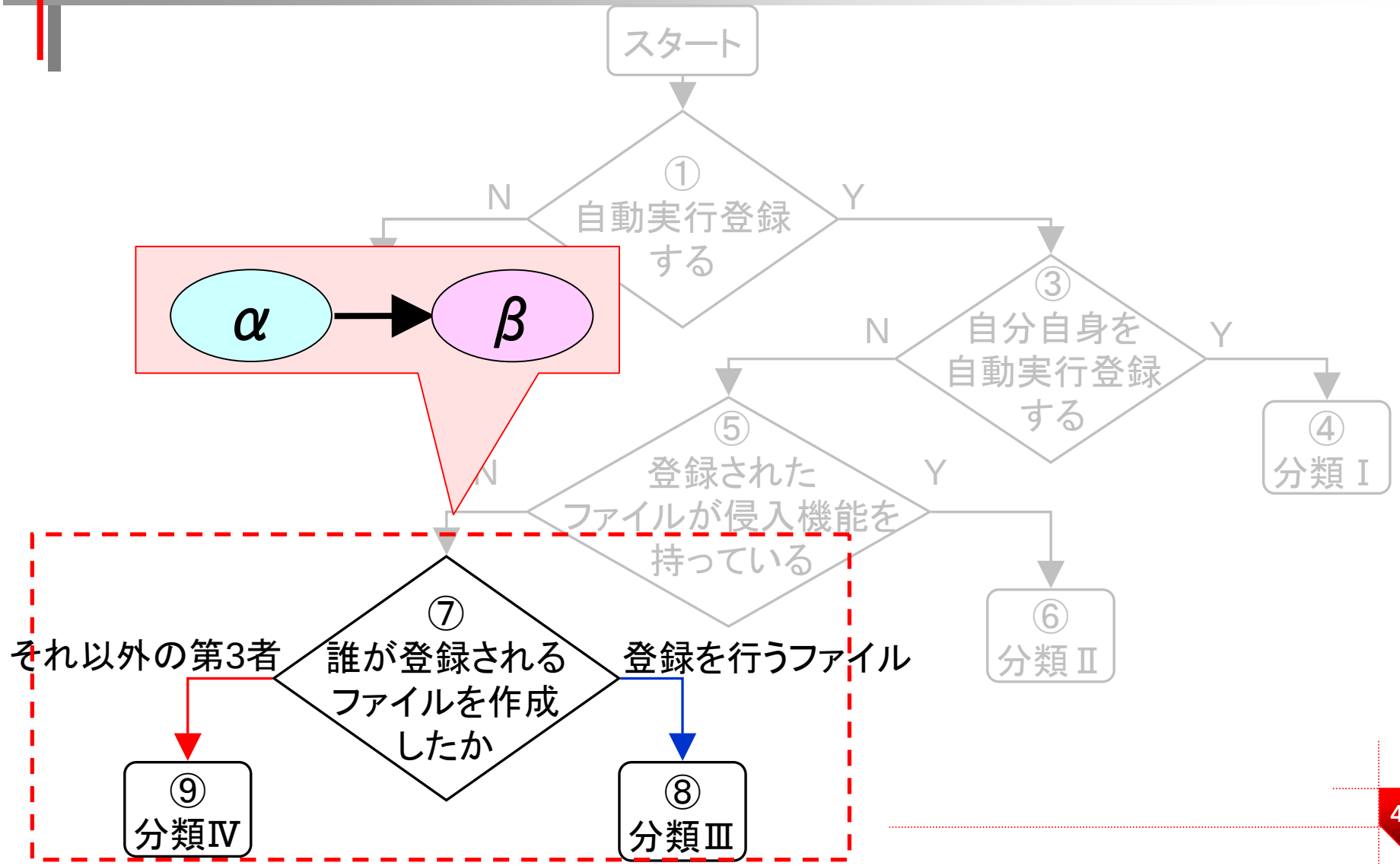
登録



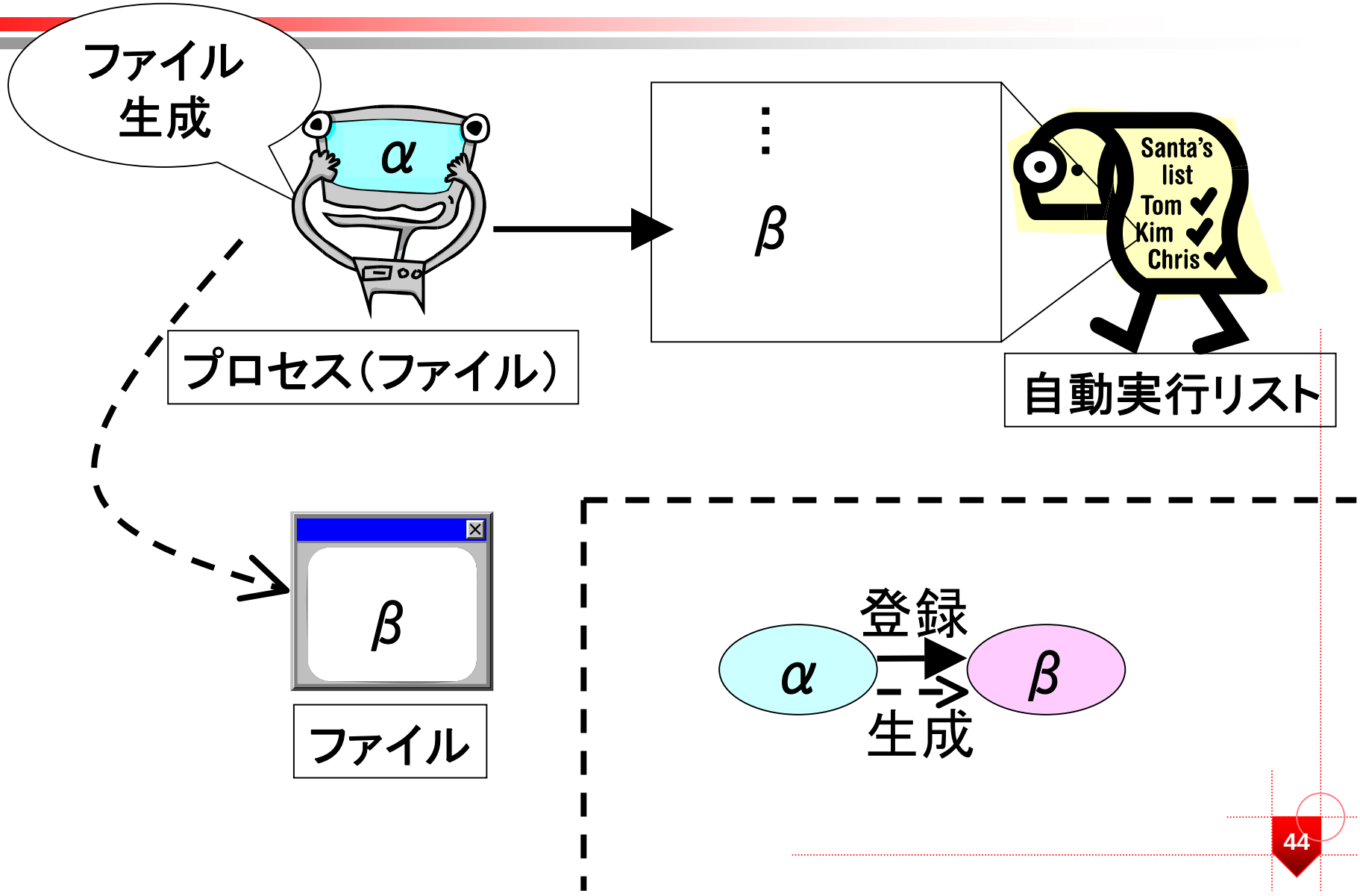
分類手順3


スタート
①
自動実行登録する
N
Y
α
β
α
β
?
⑤
登録されたファイルが侵入機能を持っている
N
Y
⑦
誰が登録されるファイルを作成したか
それ以外の第3者
登録を行うファイル
④
分類Ⅰ
⑥
分類Ⅱ
⑨
分類Ⅳ
⑧
分類Ⅲ

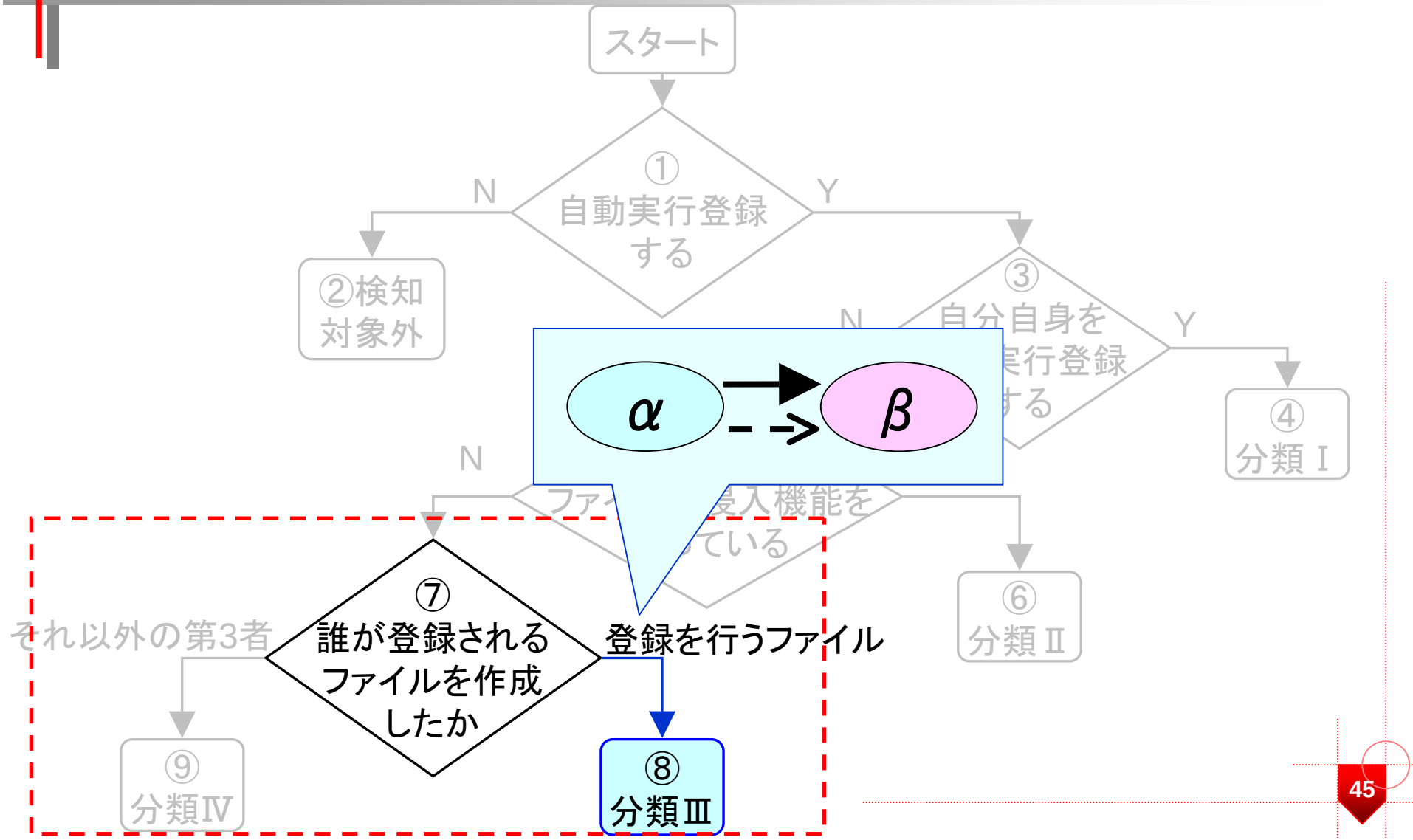
分類手順4


スタート
①
自動実行登録する
N
Y
α
β
③
自分自身を自動実行登録する
N
Y
④
分類Ⅰ
⑤
登録されたファイルが侵入機能を持っている
N
Y
⑥
分類Ⅱ
⑦
誰が登録されるファイルを作成したか
それ以外の第3者
登録を行うファイル
⑨
分類Ⅳ
⑧
分類Ⅲ

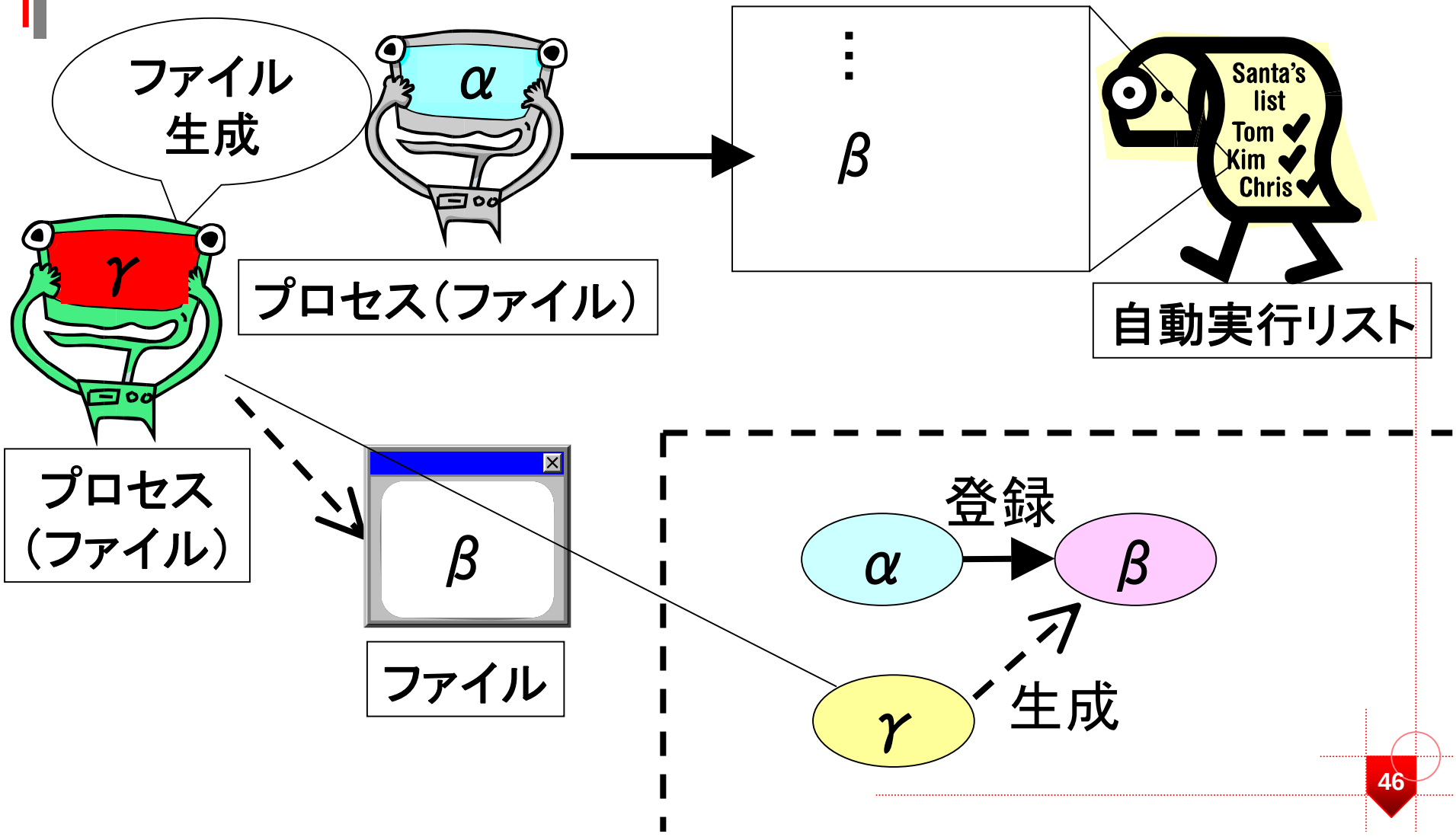
⑦の分類(分類Ⅲ)


ファイル
生成
α
プロセス(ファイル)
β
Santa's list
Tom
Kim
Chris
自動実行リスト
β
ファイル
登録
α
β
生成

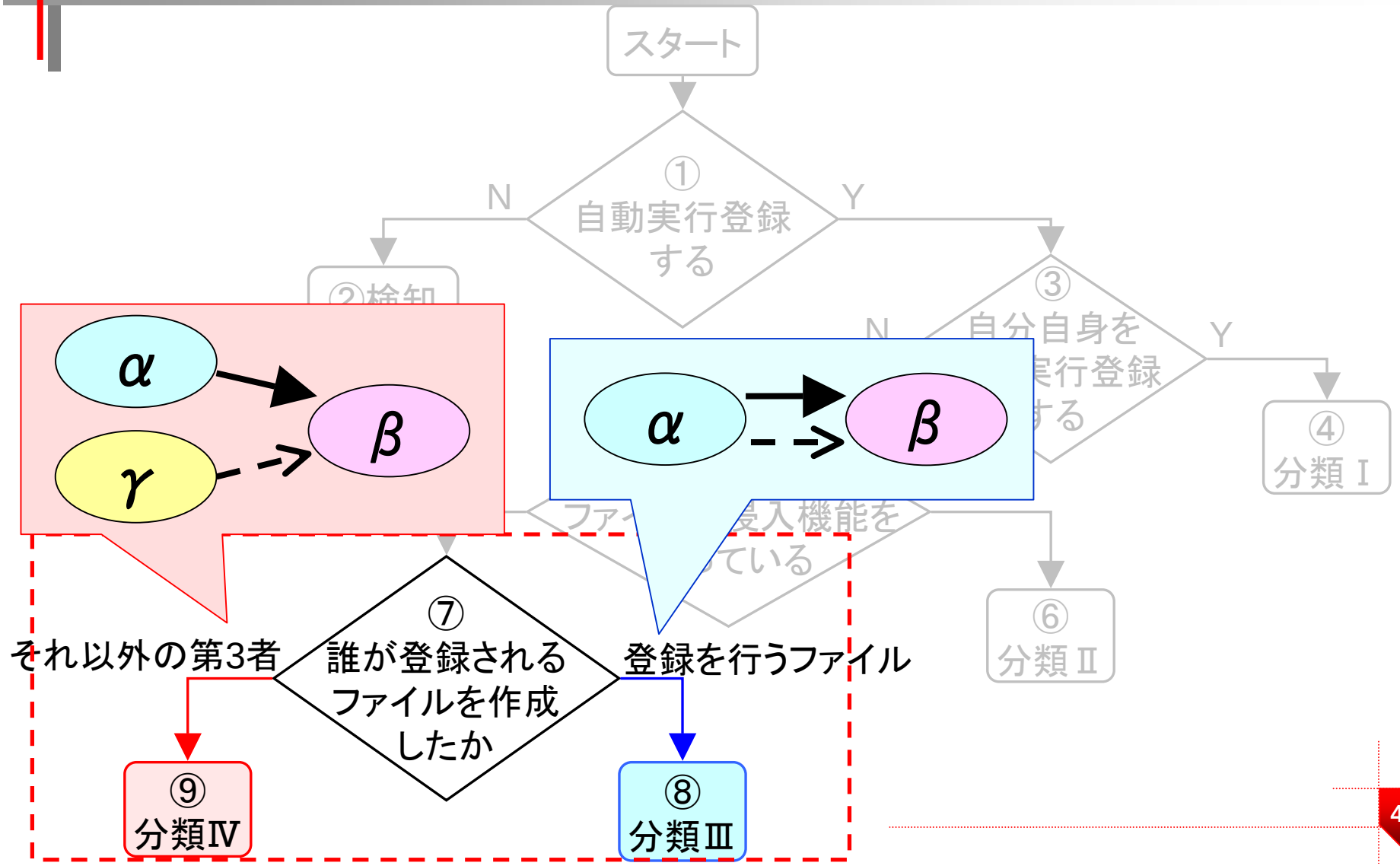
分類手順4


スタート
①
自動実行登録する
N
Y
②検知対象外
③
自分自身を
実行登録
する
N
Y
④
分類Ⅰ
α
β
N
ファ
侵入機能を
ている
⑥
分類Ⅱ
⑦
誰が登録される
ファイルを作成
したか
それ以外の第3者
登録を行うファイル
⑨
分類Ⅳ
⑧
分類Ⅲ

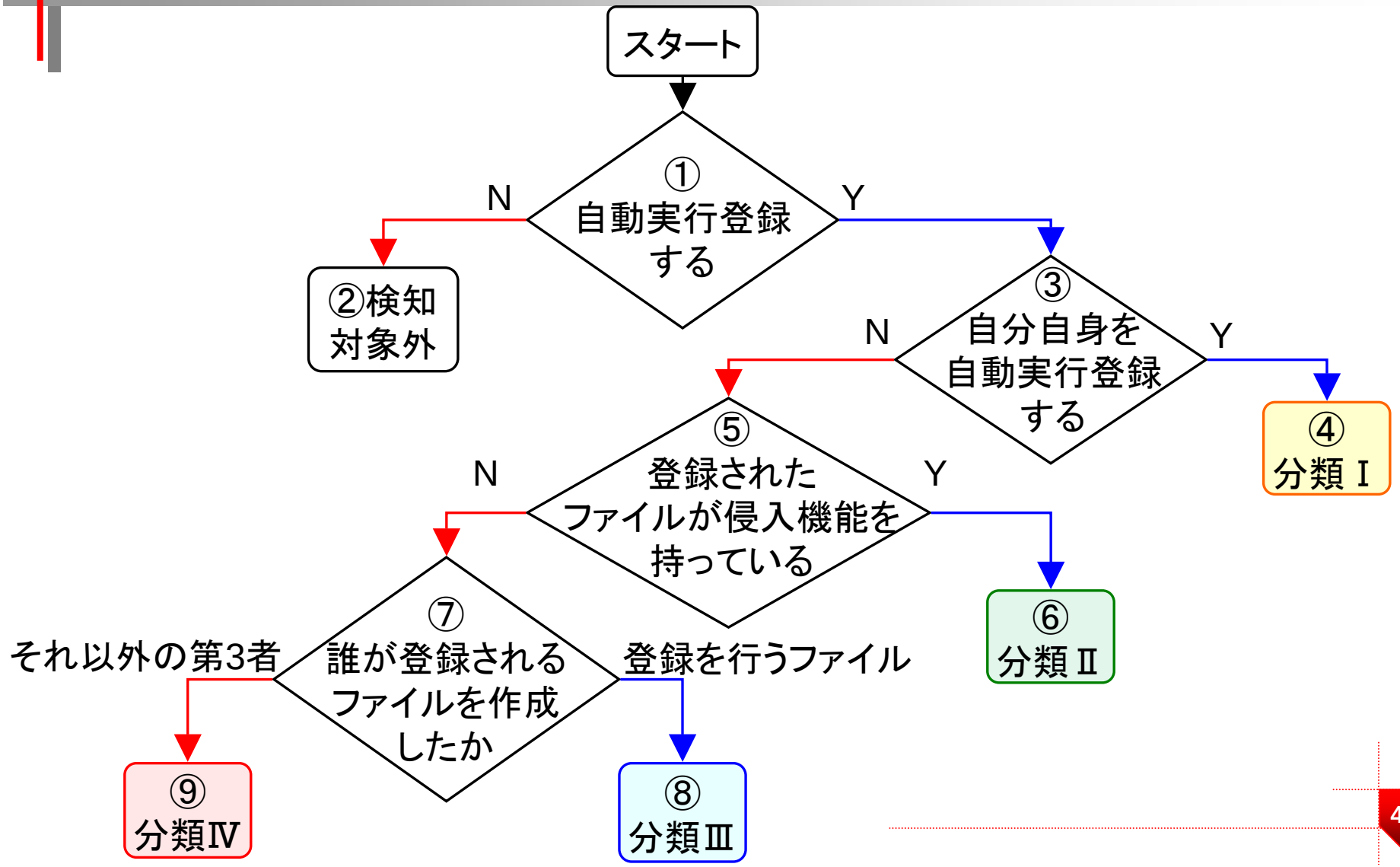
⑦の分類(分類Ⅳ)


ファイル
生成
α
γ
プロセス(ファイル)
プロセス
(ファイル)
β
ファイル
⋮
β
Santa's
list
Tom
Kim
Chris
自動実行リスト
登録
α
β
γ
生成

分類手順4


スタート
①
自動実行登録する
N
Y
②検知
③
自分自身を
実行登録
する
N
Y
④
分類Ⅰ
α
β
γ
α
β
ファ
侵入機能を
ている
⑥
分類Ⅱ
⑦
誰が登録される
ファイルを作成
したか
それ以外の第3者
登録を行うファイル
⑨
分類Ⅳ
⑧
分類Ⅲ

作成した分類図


スタート
①
自動実行登録する
N
Y
②検知対象外
③
自分自身を自動実行登録する
N
Y
④
分類Ⅰ
⑤
登録されたファイルが侵入機能を持っている
N
Y
⑥
分類Ⅱ
⑦
誰が登録されるファイルを作成したか
それ以外の第3者
登録を行うファイル
⑨
分類Ⅳ
⑧
分類Ⅲ

検証実験

Nish!gaki Laboratory

- 研究用データセットCCC DATAset 2010の検体50体に対して、提案方式に基づいて、監視ツール(ProcessMonitor, Autoruns)を用いた検証実験を行った。

- 実験環境
  - OS:Windows XP Professional SP2
  - 隔離されたローカルマシン

実験結果

Nish!gaki Laboratory

分類	合計
検知対象外	21
分類Ⅰ	1
分類Ⅱ	19
分類Ⅲ	2
分類Ⅳ	0
その他	5
	48

侵入挙動の反復性によって検知可能

連携感染によって検知可能

考察~検知対象外について~

Nish!gaki Laboratory

- 検知対象外に含まれる検体
  - 実験環境が整っていないために動作しない

  ⇒環境を整えることで分類を進められる可能性有り

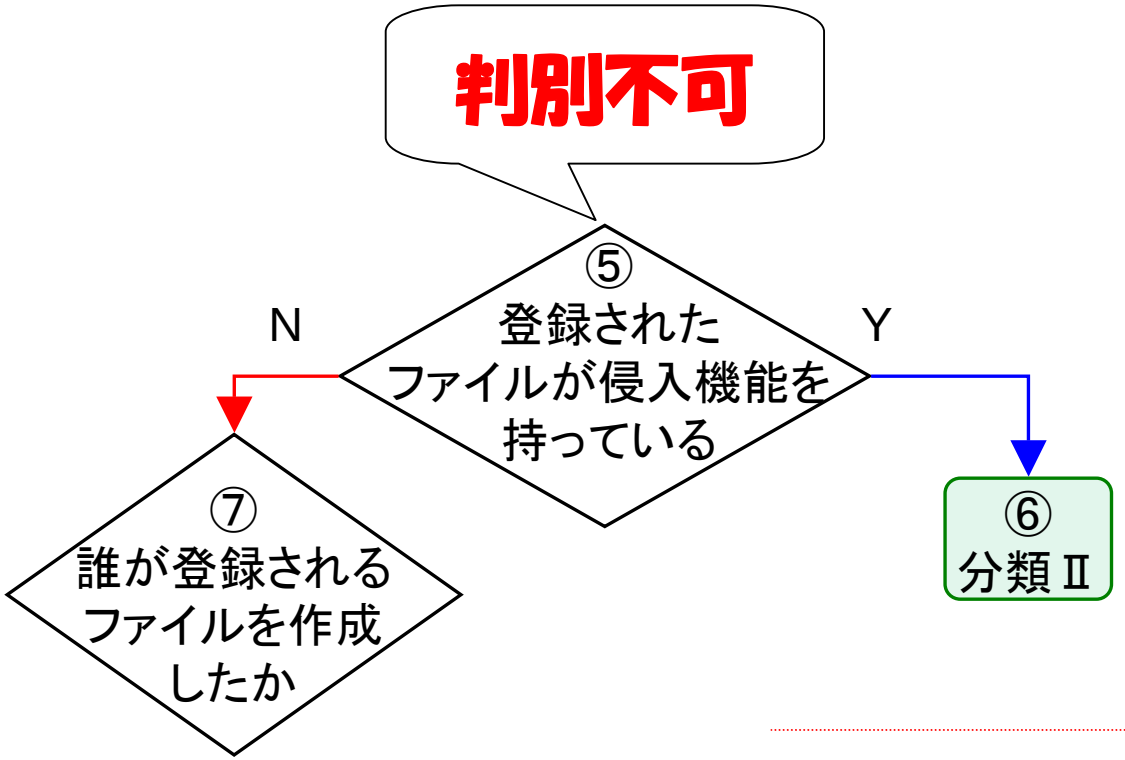
  - 自動実行登録を行わない

  ⇒自動実行登録以外の観点から，分類を検討していく必要有り

考察~その他について~

Nish!gaki Laboratory

- 自動実行登録されたファイルを
  - 何らかの方法でアクセス不可能にする.
  - 削除する



考察~その他について~

Nish!gaki Laboratory

- 自動実行登録されたファイルを
  - 何らかの方法でアクセス不可能にする.
  - 削除する

考察~その他について~

Nish!gaki Laboratory

- 自動実行登録されたファイルを
  - 何らかの方法でアクセス不可能にする.
  - 削除する

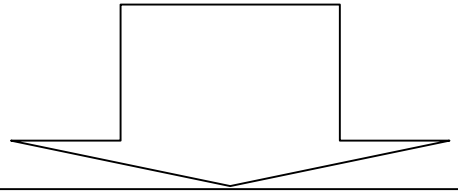
自身が生成した実行ファイルを自動実行登録していた
⇒分類Ⅱまたは分類Ⅲに含まれる.

検体を正確に分類するために，観測の精度を高める必要が有る

考察~その他について~

Nish!gaki Laboratory

- 自動実行登録されたファイルを
  - 何らかの方法でアクセス不可能にする
  - 削除する



「ファイルの存在を隠す」というという挙動

⇒特徴的な挙動

この特徴を分類の観点に組み入れることによって，新たな分類図を生成することも可能.

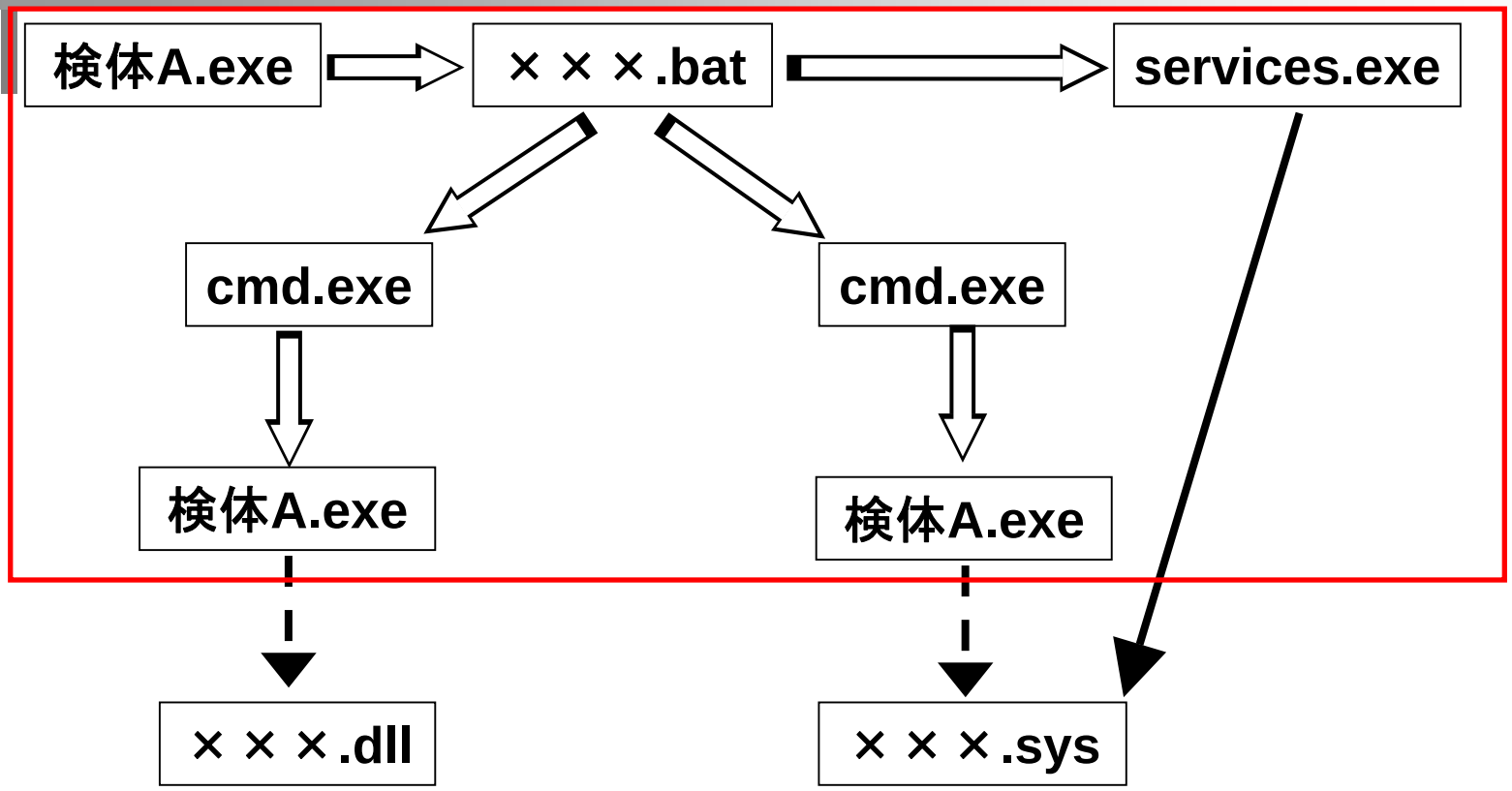
考察~分類Ⅲについて~

- 分類Ⅲのマルウェアは正規のプログラムとの区別がつかない
  - マルウェアと正規プログラムの判別をするために，さらに細かく分類を行っていく必要がある.

- 分類Ⅲに分類された2つの検体の動作を詳細に解析してみた.

分類Ⅲの検体考察（その1）

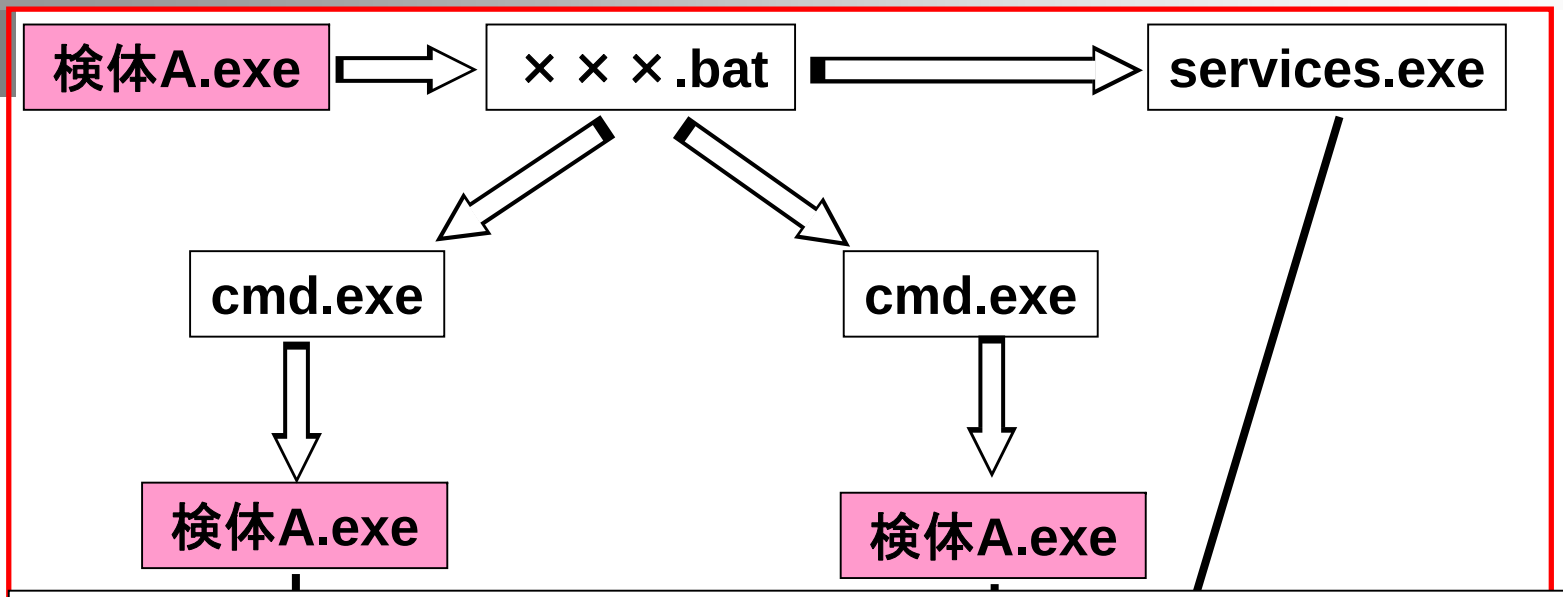
Nish!gaki Laboratory



自動実行登録
ファイル生成
生成・実行・依頼
実行ファイルのリンク

分類Ⅲの検体考察（その1）

Nish!gaki Laboratory



自分自身を再実行する動作

⇒特徴的な挙動

「実行のリンク」に注目することで，分類Ⅲのマルウェアをさらに分類することができる可能性がある

分類Ⅲの検体考察(その2)

- 特徴的な動作
  - 実行するたびに異なるファイル名の実行ファイルを生成・自動実行登録する.

「生成ファイル名」に着目することで，分類Ⅲのマルウェアを更に分類することができる可能性がある.

まとめ

Nish!gaki Laboratory

- 自動実行登録に基づくマルウェアの分類図を作成した
- マルウェアの分類図より，今後検討すべき課題を明らかにすることができた.
- 今後は分類Ⅲについて検知方式を検討していく.