

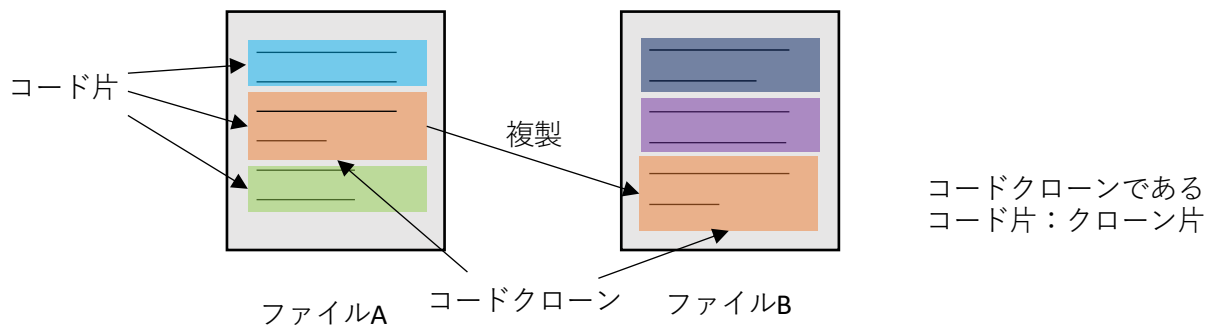
コードクローンに特化した 欠陥混入予測手法の構築

オープンソースソフトウェア工学研究室 M2

60200089 野口 耕二郎

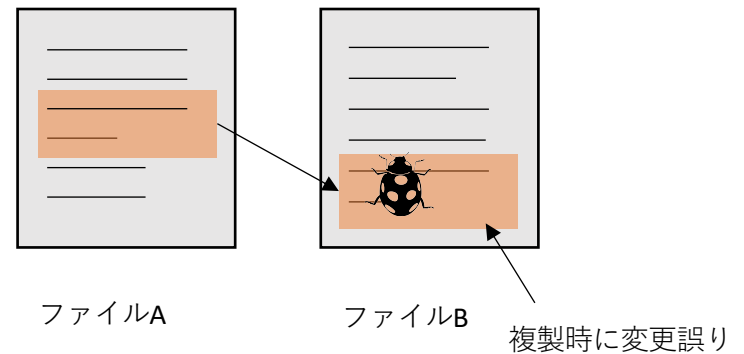
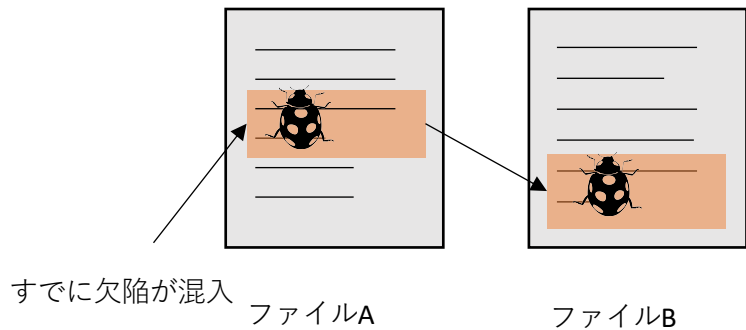
コードクローン

- 「一致」または「類似」するソースコードの断片
- 生産性を向上させるためにコードクローンの作成が頻繁に行われる
- コードクローン作成時に欠陥が複製されたり新たに混入する場合がある



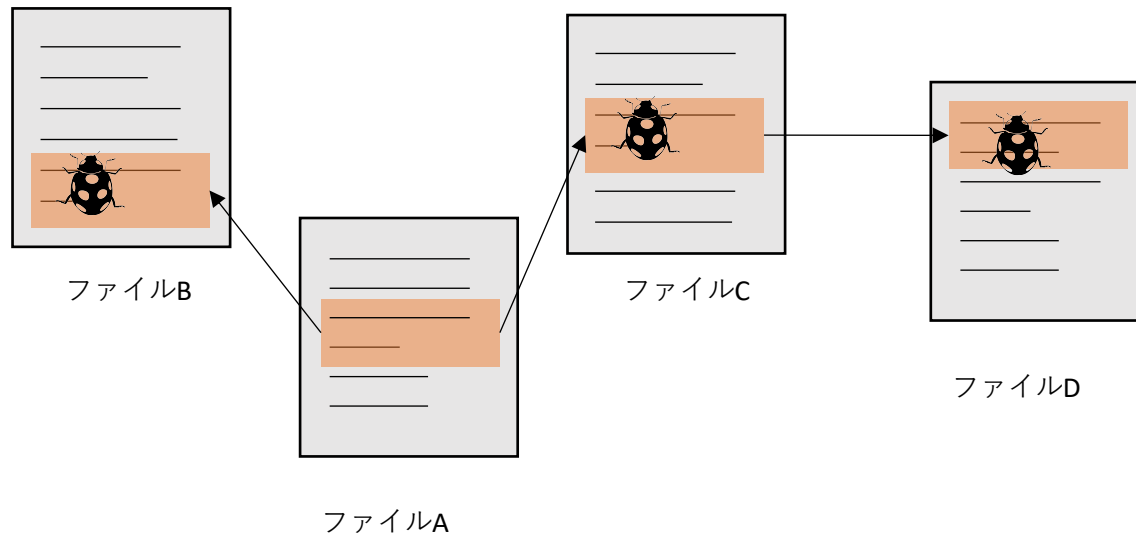
コードクローン生成時に混入する欠陥

- 欠陥が混入しているコードを欠陥が混入したまま複製（左）
- 複製時に適切な変更を加えなかったことで複製先のコードに欠陥が混入（右）



コードクローン作成時の欠陥混入の問題点

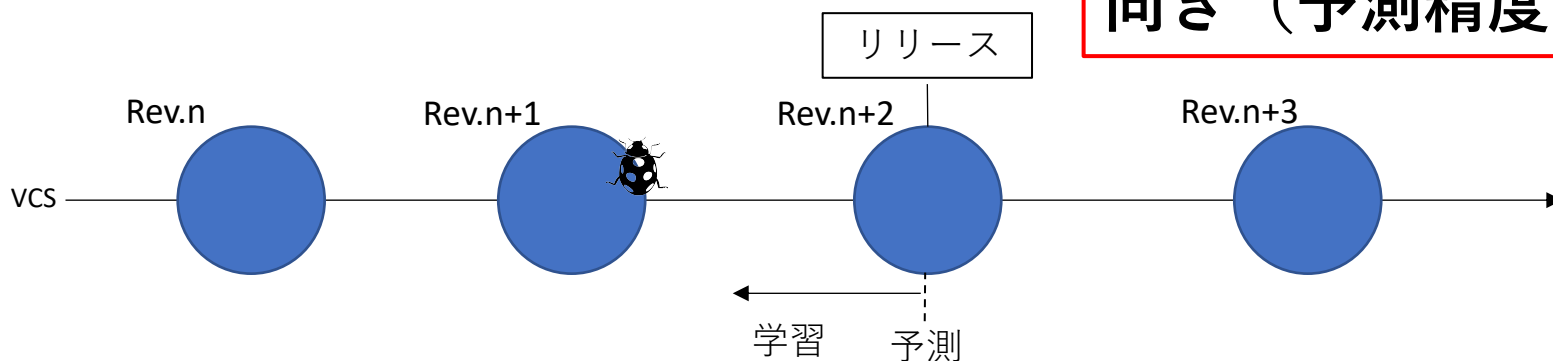
- コードクローンの作成時の**欠陥の拡散**により生産性が低下する
 - 修正すべき箇所が増加する
 - 欠陥が混入したクローン片を発見する必要がある



欠陥が広く拡散する前にできるだけ早く欠陥混入クローンを検出・除去する必要がある

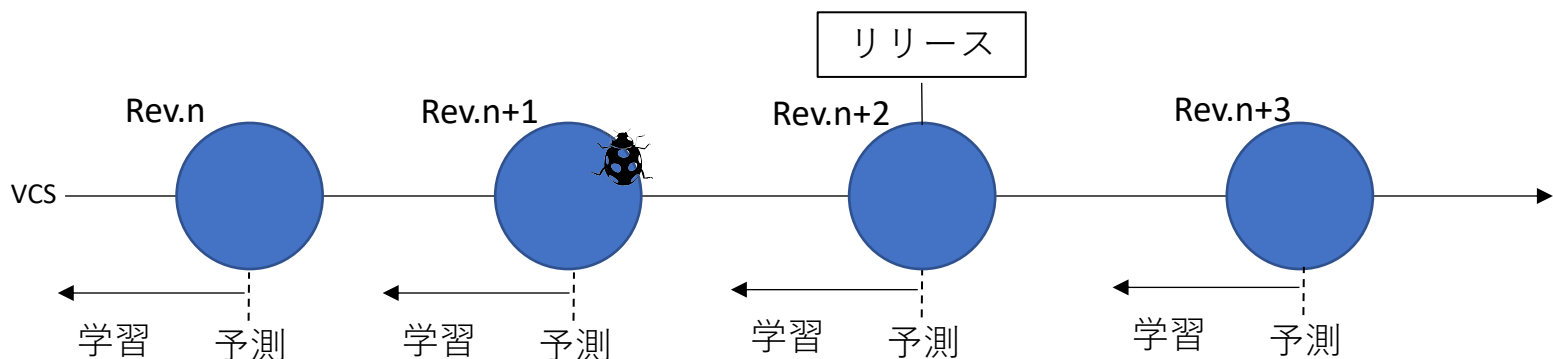
先行研究

- 従来の欠陥混入予測手法



欠陥の早期検出には不向き（予測精度重視）

- JIT欠陥混入予測手法[1][2]



欠陥の早期検出を目的

[1] Kamei, Y.; Shihab, E.; Adams, B.; Hassan, A. E.; Mockus, A.; Sinha, A. & Ubayashi, N. A large-scale empirical study of just-in-time quality assurance

IEEE Transactions on Software Engineering, 2013, 39, 757-773

[2] McIntosh, S. & Kamei, Y. Are Fix-Inducing Changes a Moving Target? A Longitudinal Case Study of Just-In-Time Defect Prediction IEEE

Transactions on Software Engineering. 2018. 44. 412-428

問題点

- JIT欠陥混入予測手法がコードクローンに混入した欠陥を検出できるか不明
 - JIT欠陥混入予測の既存研究では、欠陥の種類を区別していない
- JIT欠陥混入予測手法で検出できない欠陥混入クローンが存在する場合、手法を構築し直す必要がある

目的

1. JIT欠陥混入予測手法で欠陥混入クロンを予測できるかを明らかにする
2. コードクロンに特化した欠陥混入予測手法を構築する

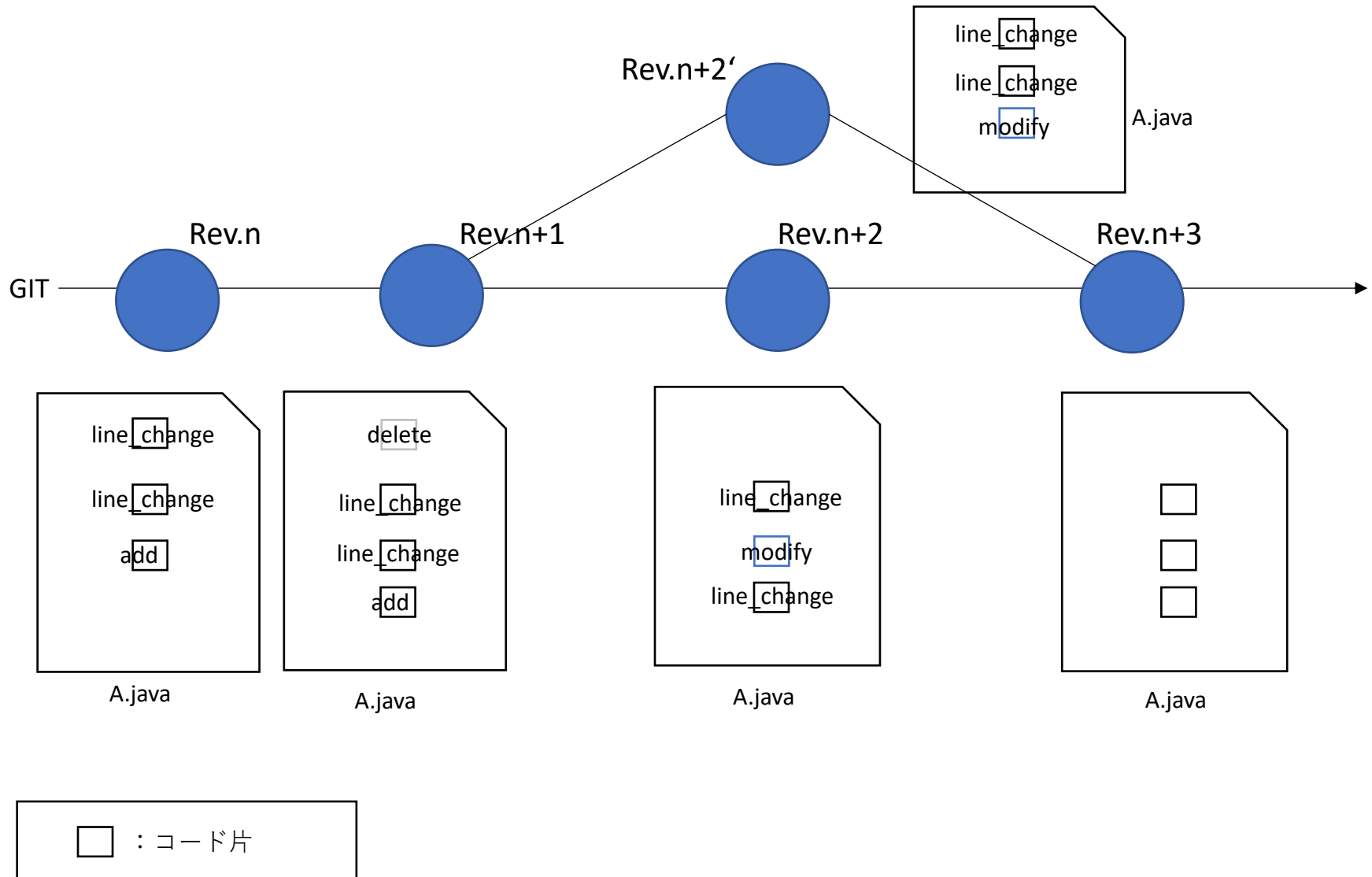
1. JIT欠陥混入予測手法を用いた欠陥混入クローンの予測

- I. 欠陥混入クローンの特定制
- II. 欠陥混入クローン作成リビジョンを目的変数として既存手法の適用
- III. 予測精度の算出

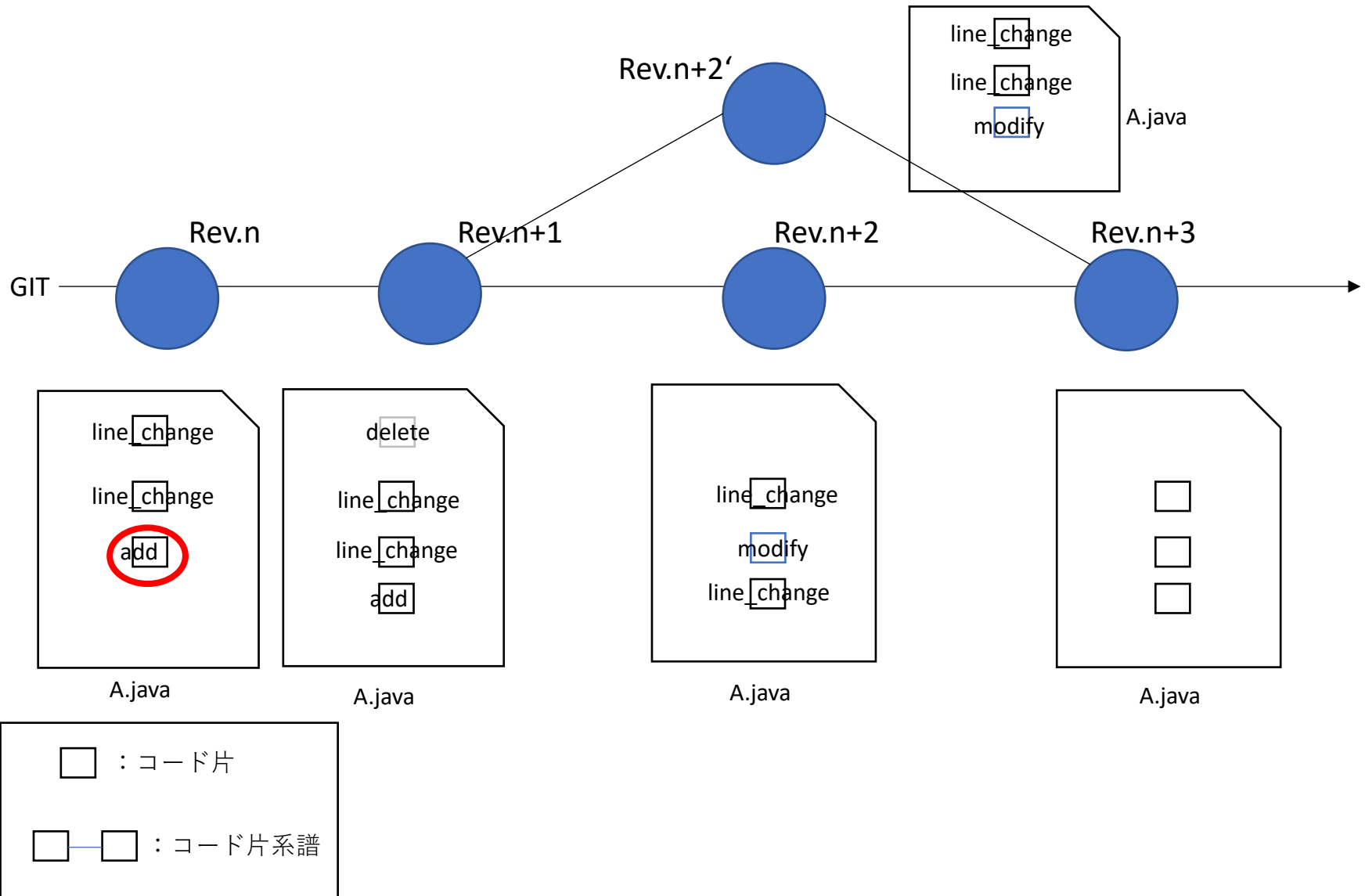
I. 欠陥混入クローンの特定

- 欠陥混入クローンの特定にはコード片系譜の抽出が必要
 - コード片系譜：コード片が変更履歴の中で加えられた変更の歴史
- 既存手法で用いているプロジェクトの版管理システム（VCS）は分散型VCSのGIT
 - 分散型VCSでのコード片系譜の抽出方法は確立されていない
 - 分散型VCSに対応したコード片系譜の抽出を実装した（本研究の貢献1）

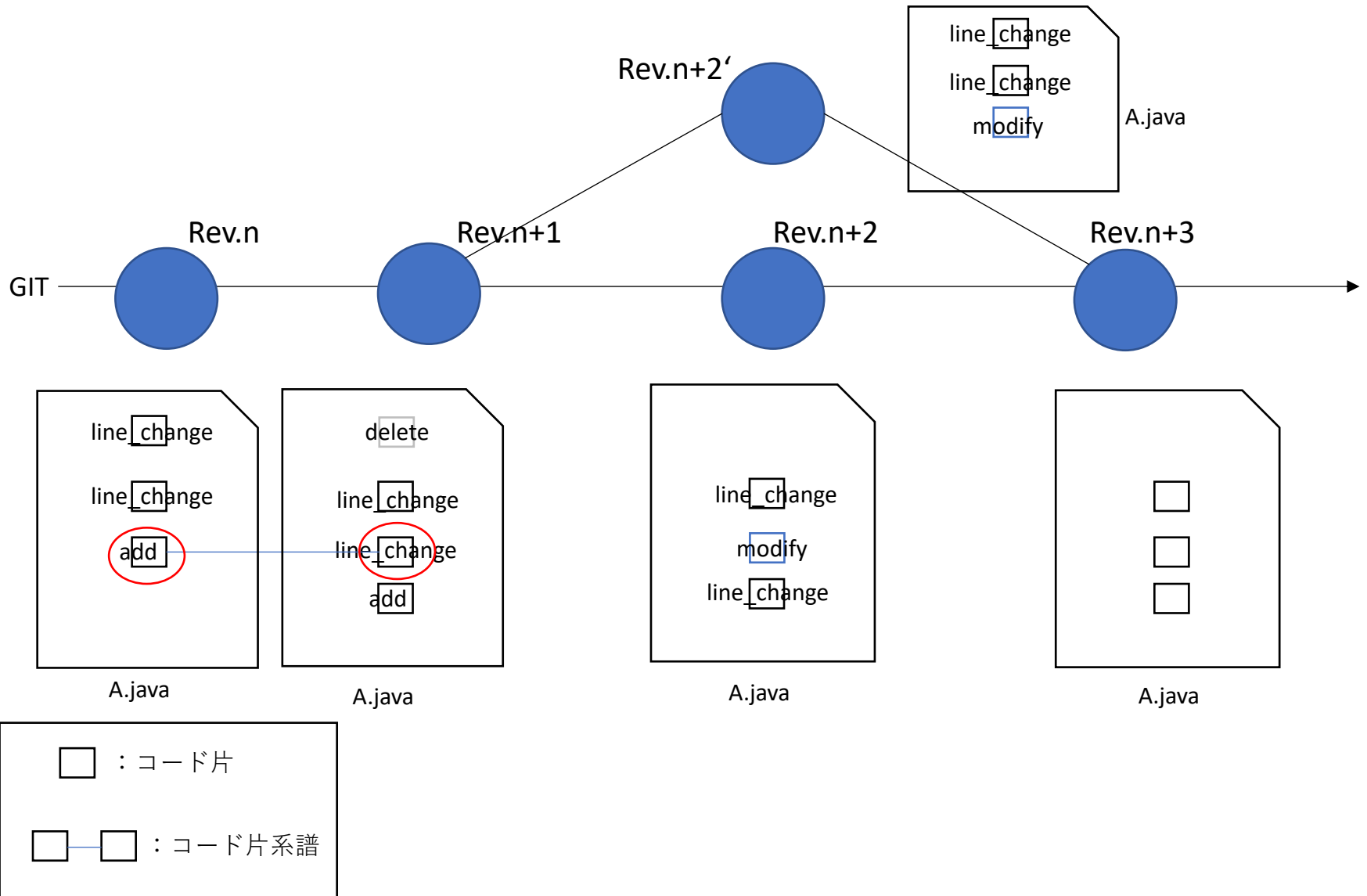
分散型VCSでのコード片系譜の抽出



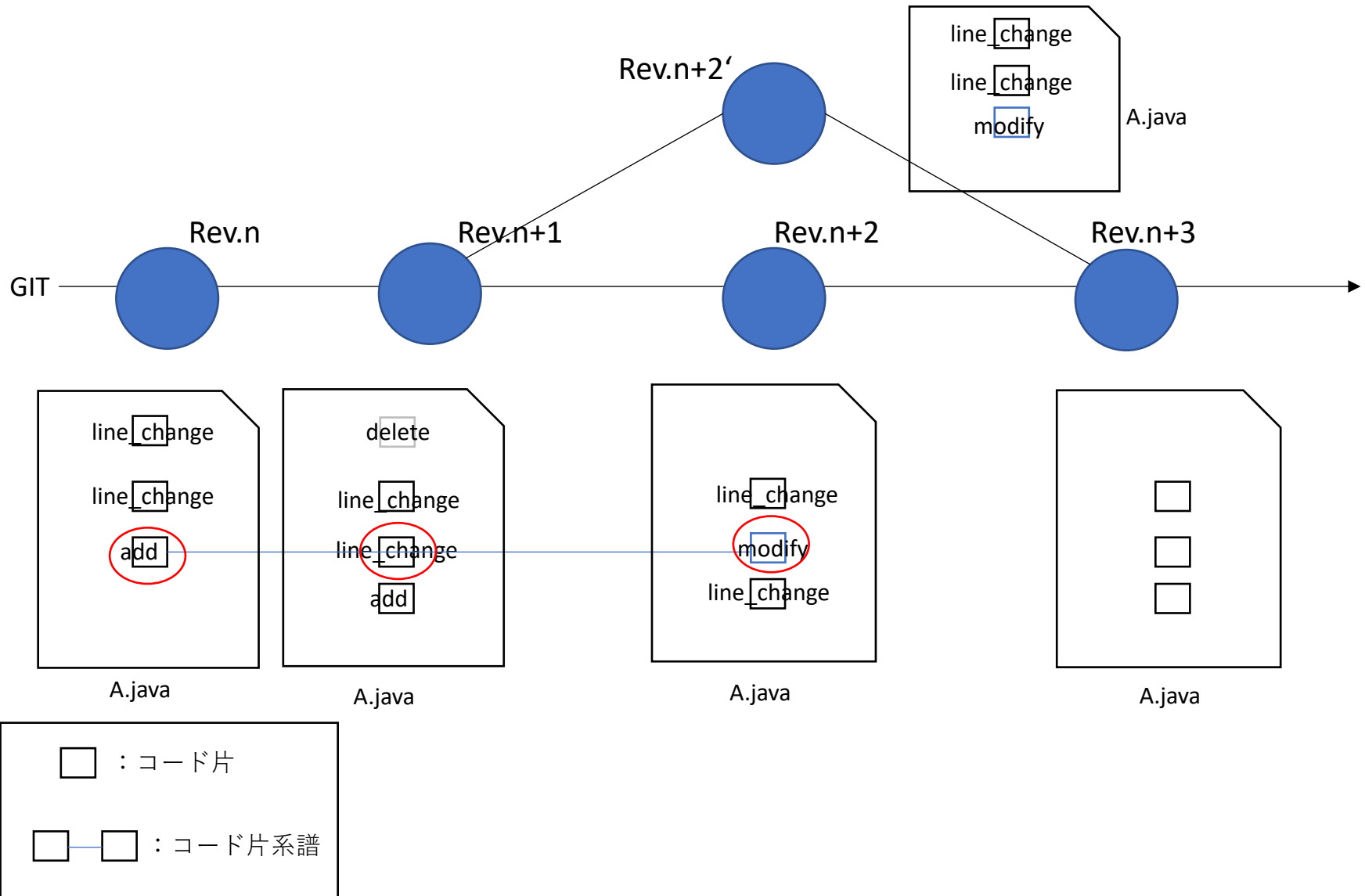
分散型VCSでのコード片系譜の抽出



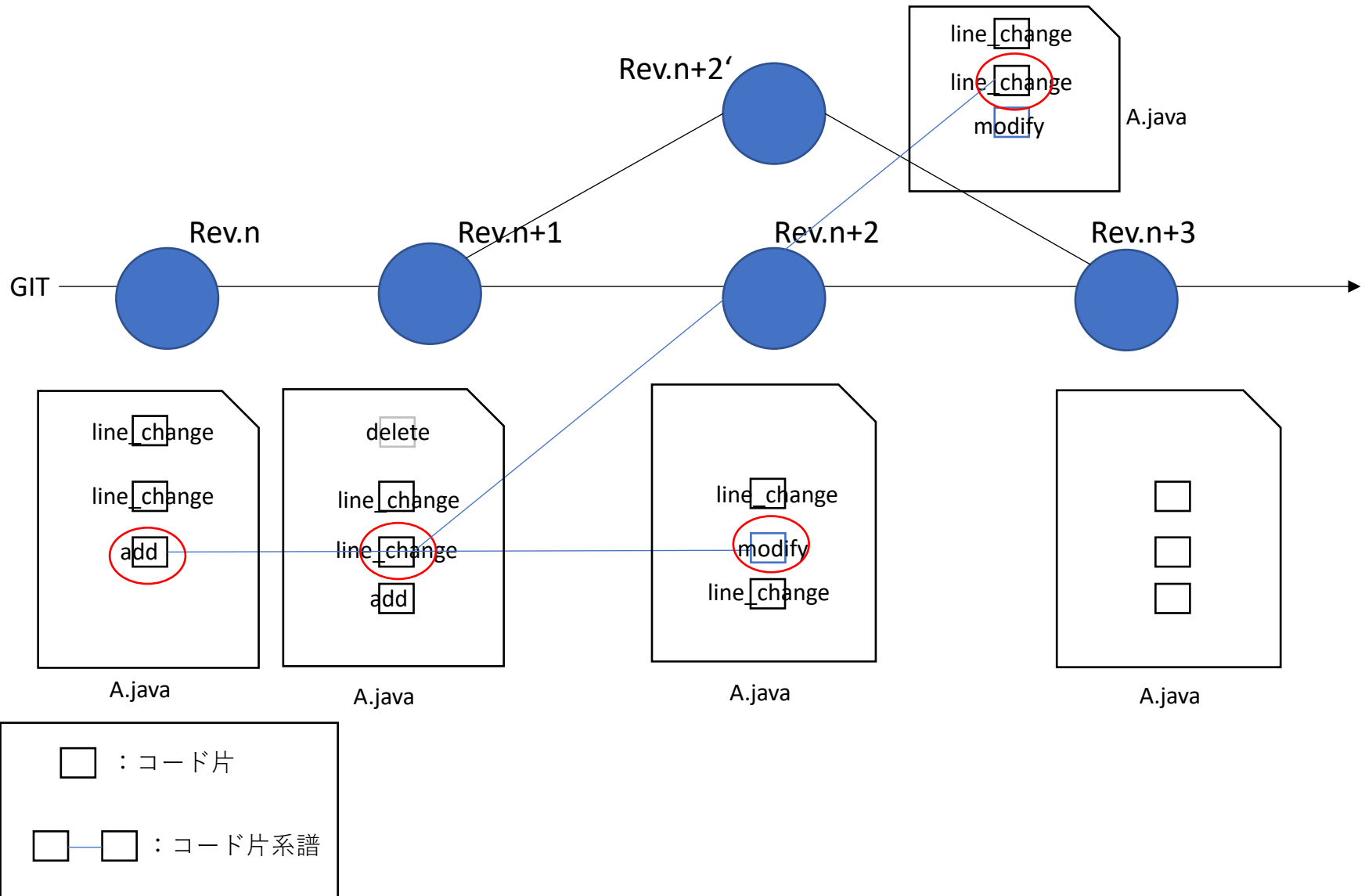
分散型VCSでのコード片系譜の抽出



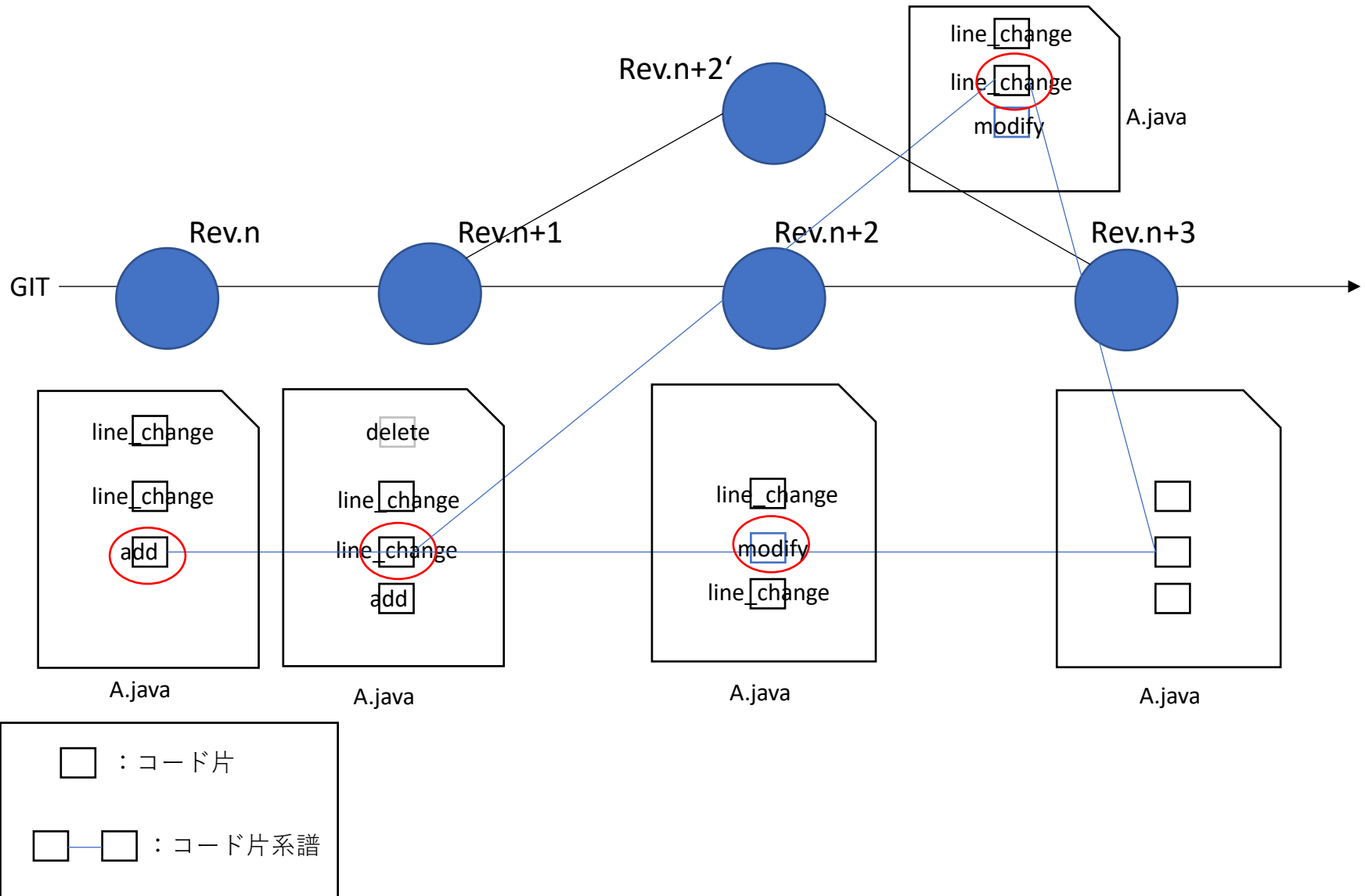
分散型VCSでのコード片系譜の抽出



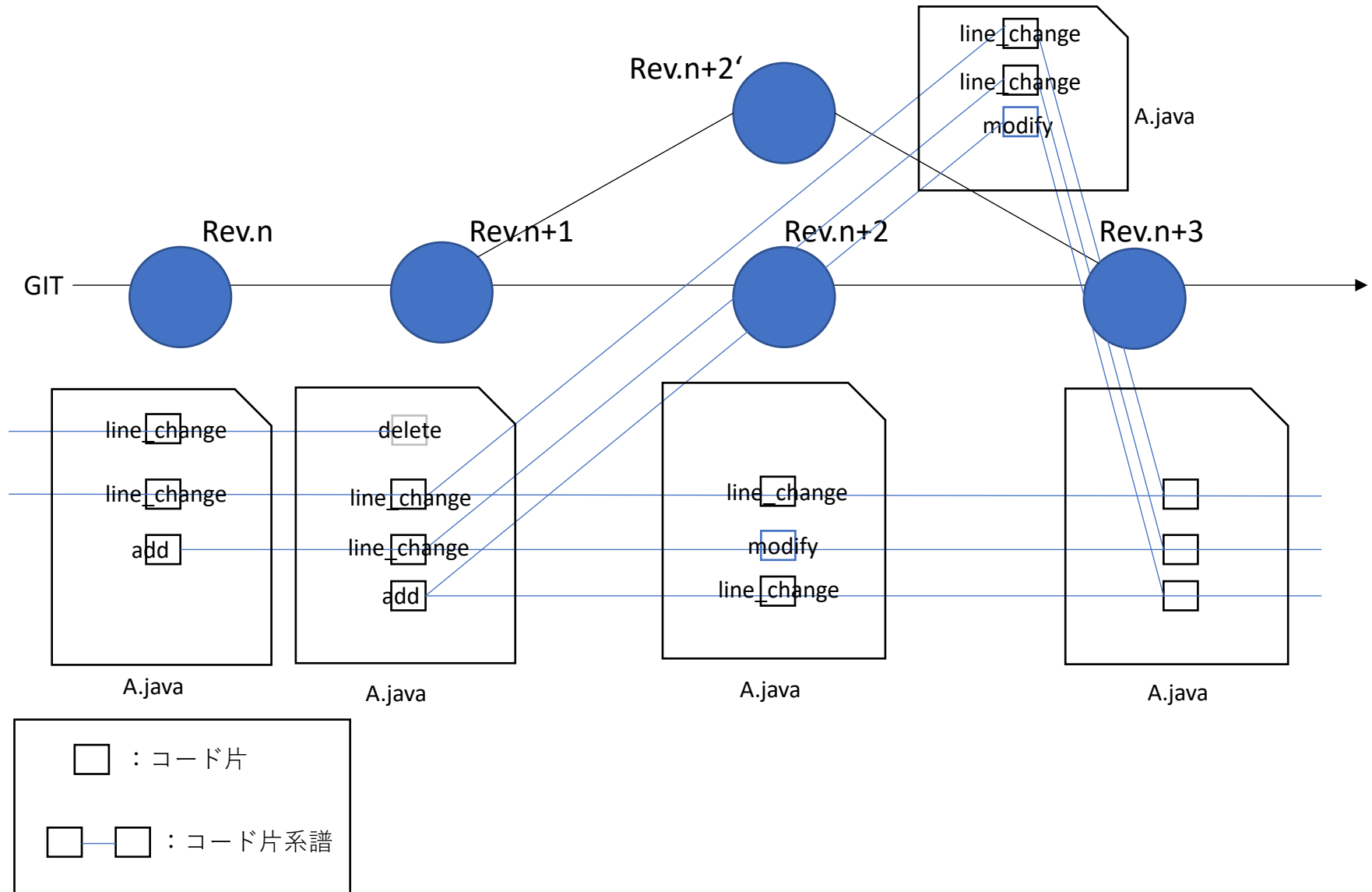
分散型VCSでのコード片系譜の抽出



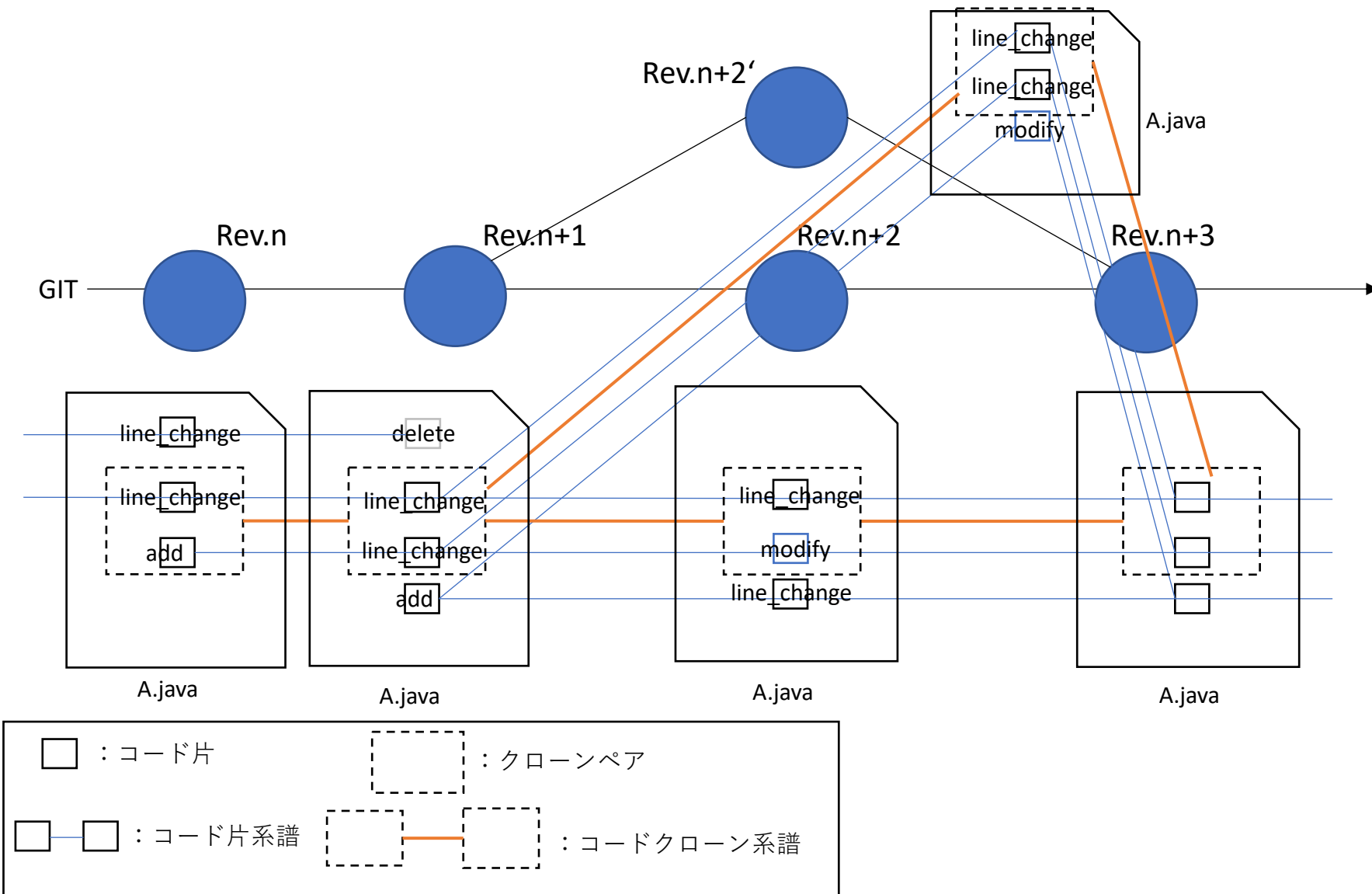
分散型VCSでのコード片系譜の抽出



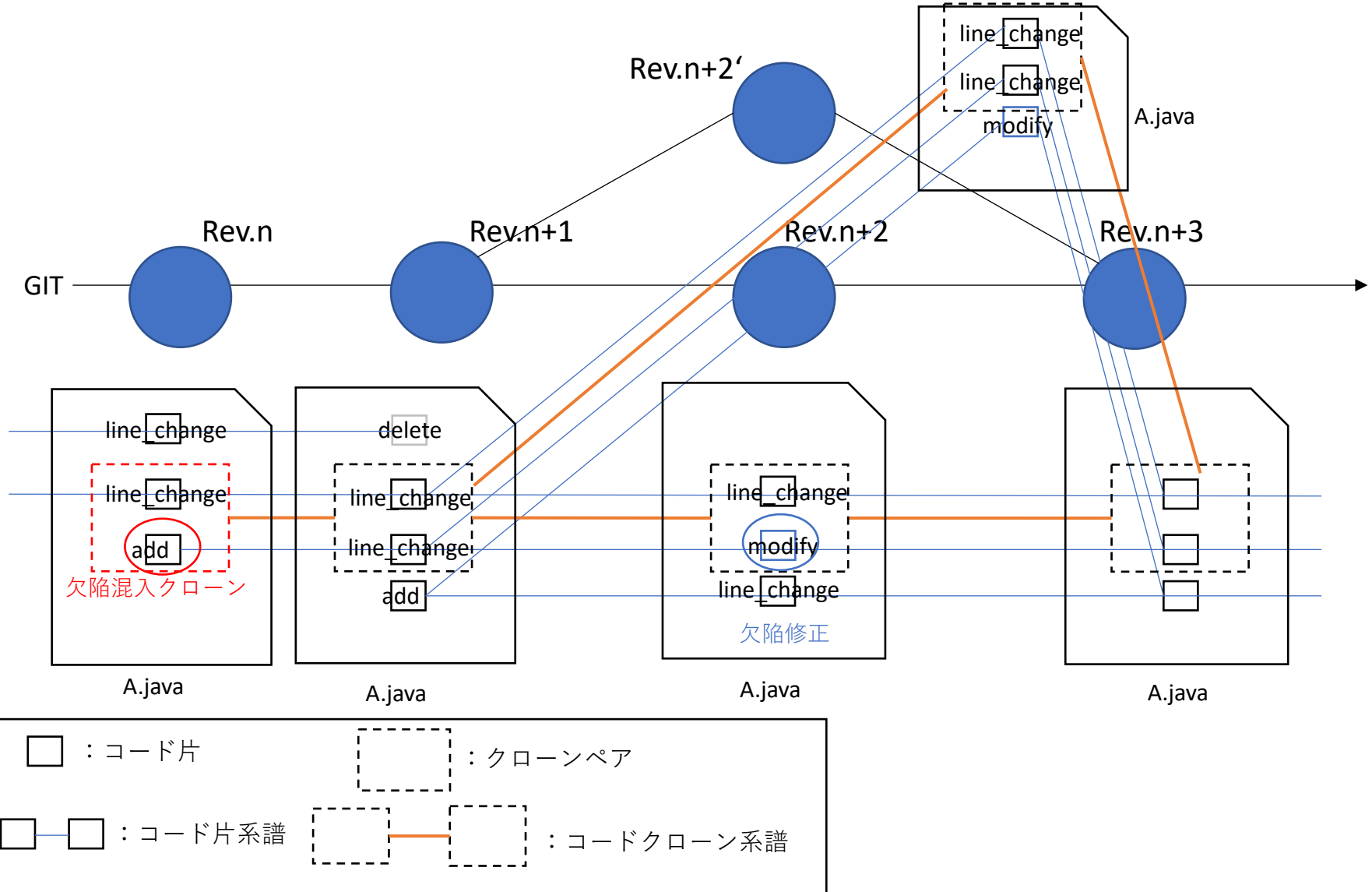
分散型VCSでのコード片系譜の抽出



コードクローン系譜の抽出



欠陥混入クローンの特定



II. 既存手法の適用

- 予測アルゴリズム

- ロジスティック回帰

- $$p = \frac{1}{1 + \exp\{-(a_1 \cdot x_1 + a_2 \cdot x_2 + \dots + a_{17} \cdot x_{17} + b)\}}$$
 - p : 目的変数に設定した事象が発生する確率
 - $a_1, a_2, \dots, a_{17}$: 偏回帰係数
 - $x_1, x_2, \dots, x_{17}$: **説明変数17種類** (次スライド)
 - b : 定数

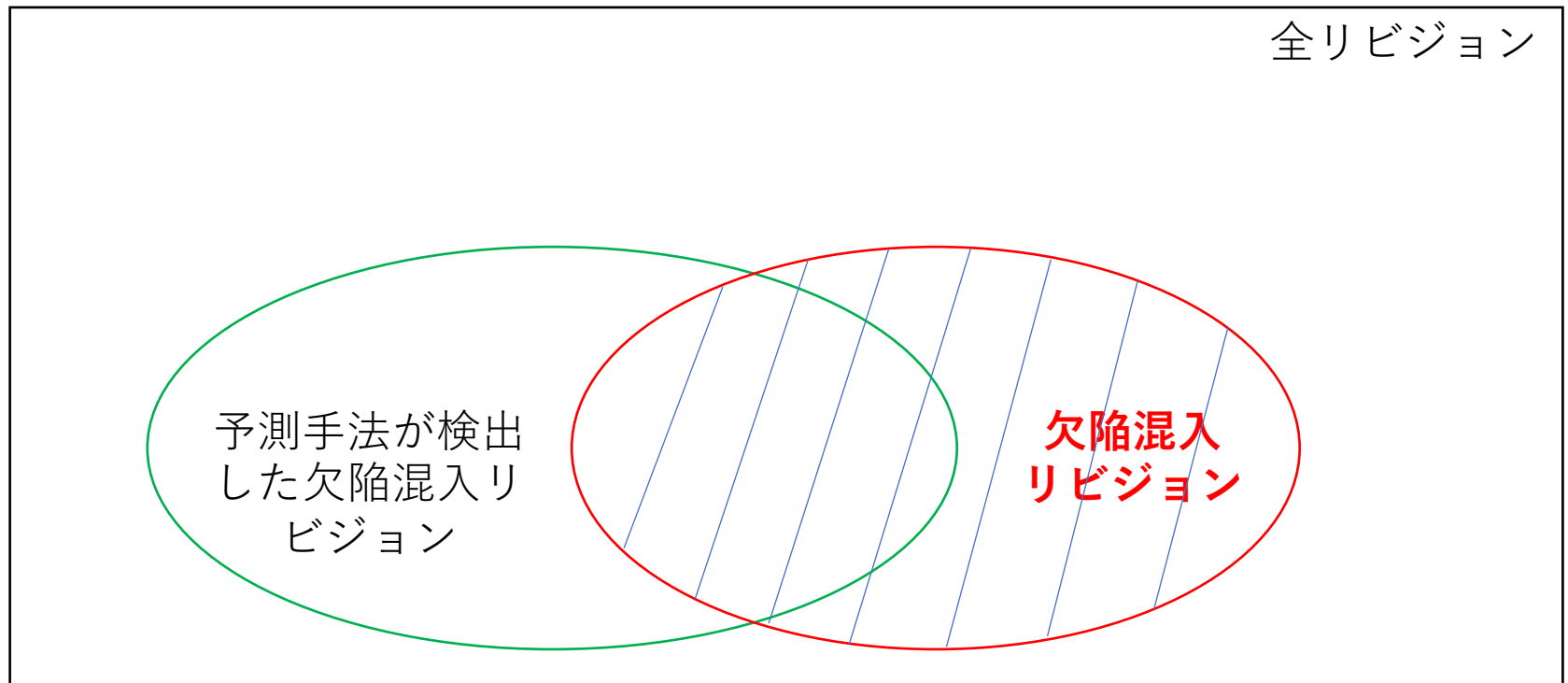
- $p \geq 0.5 \rightarrow$ 欠陥混入 $p < 0.5 \rightarrow$ 欠陥未混入

II. 既存手法の適用：説明変数

- 5種類，17項目のメトリクスを説明変数とする
 - 変更規模 x 2種類（例：追加行数）
 - 変更の拡散度合い x 4種類（例：変更ファイル数）
 - 歴史 x 3種類（例：ファイル歴代変更回数）
 - 開発者の経験 x 4種類（例：開発者が加えた変更回数）
 - コードレビュー x 4種類（例：レビューアー数）

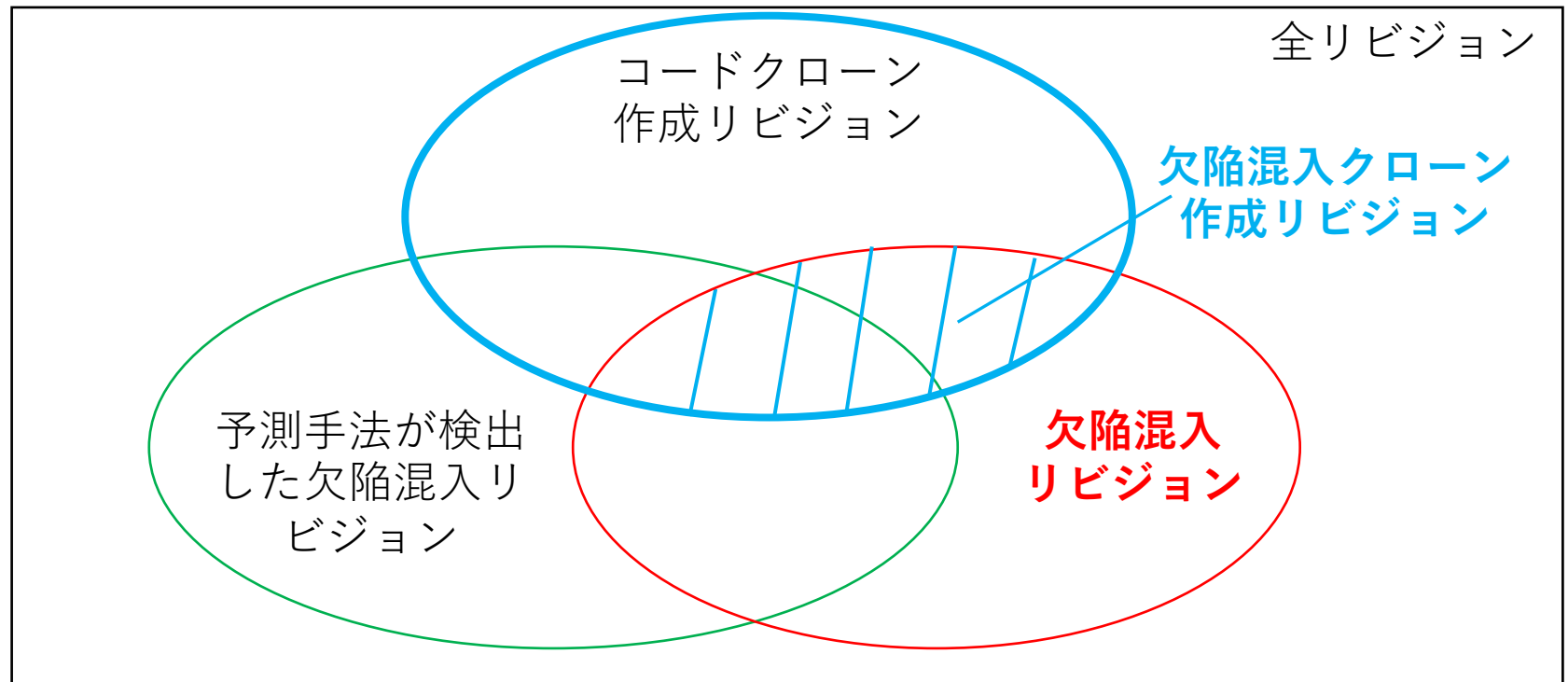
II. 既存手法の適用： 先行研究の予測対象

- 先行研究では、**欠陥混入リビジョン**を予測対象としている



II. 既存手法の適用： 本研究の予測対象

- 欠陥混入クローン作成リビジョンを予測対象とする

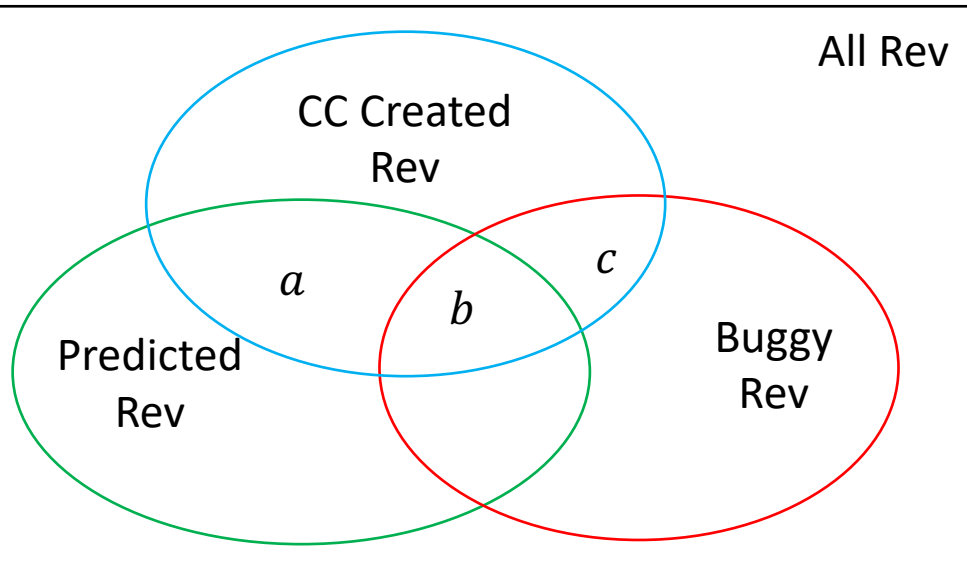


III. 予測精度の算出

- a , b , c をもとに再現率, 適合率を算出する

- 再現率 : $\frac{|b|}{|b \cup c|}$

- 適合率 : $\frac{|b|}{|b \cup a|}$



$b \cup c$: 欠陥混入クローン作成リビジョン

b : 既存手法で検出できた欠陥混入クローン作成リビジョン

c : 既存手法で検出できなかった欠陥混入クローン作成リビジョン

a : 既存手法が誤って検出したクローン生成リビジョン

データセット

- QTプロジェクト（29プロジェクト）
 - 合計リビジョン数： 27589
- Openstackプロジェクト（5プロジェクト）
 - 合計リビジョン数： 13209
 - 予備実験（cinderプロジェクト）

プロジェクト	期間	リビジョン数	欠陥混入リビジョン数	クローン作成リビジョン数	欠陥混入クローン
cinder	2011/11– 2014/3	2,005	343	183	9

予備実験：欠陥混入クローンの予測

- データを訓練データに80%，テストデータに20%振り分けて欠陥を予測
 - cinderプロジェクトに含まれる欠陥混入クローンの件数が9件と少なかったため
 - 本実験では，サブプロジェクトを結合して予測・10交差検証を実施する予定
- テストデータに含まれる欠陥混入クローンの内，予測できたものを集計した結果，JIT欠陥予測手法では欠陥混入クローンを予測できなかった

テストデータに含まれる欠陥混入クローン	予測できた欠陥混入クローン
2	0

今後の予定

- 研究目的1：
JIT欠陥予測手法による欠陥混入クローンのパ測
- 先行研究と同規模にデータを増やして精度を算出
(12月中旬に終了予定)
- 研究目的2：
コードクローンに特化した欠陥予測手法の構築
- JIT欠陥予測手法で検出できない欠陥混入クローンの特
徴を分析し、予測モデルを修正する
(1月中旬に終了予定)