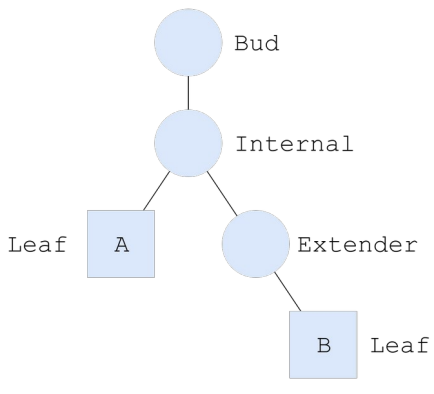
暗号通貨Tezos向けストレージシステムPlebeiaにおけるディスクを介したデータ永続化処理のF*による形式検証

伴野 良太郎[*1]・佐藤 聡太[*1]・古瀬 淳[*2]・末永 幸平[*1]・五十嵐 淳[*1]
[*1] 京都大学　[*2] DaiLambda, Inc.

暗号通貨における形式検証

- 暗号通貨システムにおけるバグは金銭的に大きな損害を生む（e.g., The DAO事件）
- 鉄道・航空機開発などではバグの混入を防ぐために従来から形式検証を使用
- 同様にバグが致命的な暗号通貨においても形式検証が有用



Plebeia treeの例
通常のMPTと異なり、内部ノードを複数種類（特にInternalとExtender）に分けることで、ディスク効率を高める

Plebeia

- OCamlにより実装されている暗号通貨Tezos向けのkey-valueストア
- Tezosのブロックチェーンに関わるデータの保持に特化
- データをMerkle Patricia Tree（MPT）の変種（Plebeia tree）により保持
  - Tezosが現在使用しているストレージシステム（Irmin2）よりもディスク効率がよい

F*

- 篩型・エフェクトなどのプログラム検証用機能を備えた関数型プログラミング言語
- OCamlとよく似た文法を持つ
- バックエンドにSMTソルバであるZ3を使用し、半自動的なプログラム検証が可能
- 等価なOCamlコードを抽出可能

補題「関数lengthの呼び出しは常に0以上」を定義

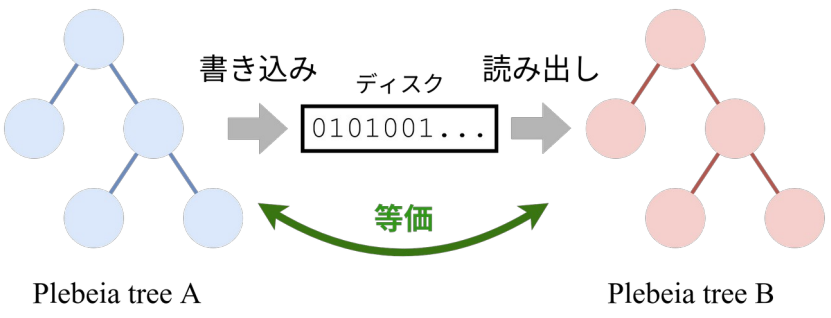
補題の証明を関数定義として記述
構造的帰納法による証明は再帰関数になる

```
val length_gte_0 : l:list ℤ → Lemma
  (requires (true))
  (ensures (length l ≥ 0))
let rec length_gte_0 l =
  match l with
  | [] → ()
  | x :: xs → length_gte_0 xs
```

リストの長さを返すlength関数に関する検証の例

検証した性質

- 命題「木Aをディスクに書き込み、次いでディスクから読み込み木Bを再構築したとき、AとBが等価である」を示す
  - 書き込み・読み込み時にはビット演算などを用いた木のシリアライズ・デシリアライズ処理を行う必要があるため、これらの処理の対応を示す
  - シリアライズ後のバイナリに含まれる、子ノードを指すポインタが正しいことを示す



検証すべき性質のイメージ

検証の方針

- 大まかには、木Aについて構造的帰納法を適用する
  - 葉ノードについては、書き込み・読み込みが対応することをPlebeiaの処理に合わせ証明
  - 内部ノードnについては、nの子ノードの書き込み・読み込み処理の対応を帰納法の仮定より導出した後、その対応がnの書き込みをまたいでも保たれることを示す

検証における問題点とその対策

- **問題**：ディスクを参照のリストとして単純にモデル化すると、ディスクへの書き込みが意図せず他領域を上書きしないことの保証が困難
  - **対策**：ディスクのモデルとしてLowStar.Buffer [Protzenko他 2018]を使用
    LowStar.BufferはC言語の配列をモデル化しており、配列の各セルの分離に関わる補題が充実
- **問題**：F*では純粋かつ全域な関数のみを証明中で使用できるが、読み込み・書き込み関数は次の理由からこれを満たさない
  - （純粋性）読み込み・書き込み関数はディスクへのアクセスを伴うため、純粋関数ではない
    - **対策**：ディスクをヒープで表し、そのストアを明示的に引数・戻り値に含めた新しい関数を定義
      元々の関数との等価性は別途F*上で証明する
  - （全域性）Plebeiaに搭載されたノード遅延読み込み機能のために、再帰的な木読み込み関数の停止性を静的に確認できず、全域性を保証できない
    - **対策**：関数の引数に自然数fuelを導入し、再帰呼び出し毎にこの値を1減らす

実装と実験

- 要したF*コード：約17,000行
- 検証実行にかかる時間：約1時間
- コード抽出を行って得られたOCamlコードは、検証前の実装と同程度の速度で動作した

検証前のOCaml実装は約500行

Plebeiaのテストの通過にかかる時間
（単位：秒、試行数10）

元々の実装	我々の実装
139 ± 2	129 ± 6

今後の方針

- Plebeiaの最新版への追従
- [Sato 修論 '21]との統合
  - SatoはPlebeia treeに対する操作がkey-valueストア操作として正しく実装されていることをF*で検証
- より複雑な読み込み関数の検証