将棋プログラム用ライブラリ libshogi

藤井宏行[†]　瀧川健太[†]　高田浩生[‡]

[†]香川高等専門学校詫間キャンパス 電子システム工学科 藤井研究室 〒769-1192 香川県三豊市詫間町香田551

[‡]ティーソフトウェア　〒763-0095 香川県丸亀市垂水町 310-6

１．はじめに

将棋プログラムは人工知能について学習する上で良い教材となります．香川高専詫間キャンパス藤井研究室（以下，藤井研）では将棋を素材に，学生が探索などのプログラミング手法について学習しています．この学習において問題となったのが，将棋の対局をプログラムで表現することの難しさでした．ソースコードが公開されている数多くの著名な将棋プログラム，ライブラリはどれも英知の結晶であり，丹念に読むことにより素晴らしい知見を得ることができます．一方でそれらは，人を超えることを目標に磨き抜かれてきたプログラムでもあります．文法のみの知識で簡単に理解できるものではありません．

藤井研では，C++ の基本的な文法を知っている学生が短期間で将棋プログラムを書き始めるところまで到達できることを目標に，学習用の将棋ライブラリの開発を行っています．

前回（第25回）までに完成していた，基本データ構造と CSA 通信プロトコルによる通信機能に加えて，将棋の合法手生成機能を実装しました．これにより，ルール通りに将棋を指すための最低限の機能が用意できました．大会終了後ソースコードを公開する予定です．

２．概要

図-1 に libshogi の階層モデルを示します．

最も重要なのは，探索，定跡，学習などの戦略を司る部分です．ライブラリは，プログラマが大きな制約を感じることなく，戦略部分のプログラミングに集中できるよう考慮されたものであるべきです．

libshogi は網掛けされた機能を提供します．Foudation 層では，アトミック操作，ロック，スレッド，リスト，などの基本データ構造と操作，Game 層では，ルール通り指すための将棋機能，外部プログラムとの通信機能などを提供します．

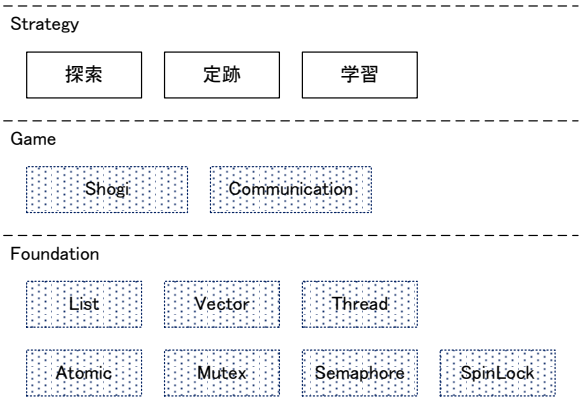

図-1　将棋プログラムの階層モデル

libshogi は Boost や STL などを使用せずプリミティブなライブラリ API のみで実装しています．また，教材を前提とするので，速度より簡明さを優先しています（Foundation 層の機能はコメントを含め合計 2000 行程度で実装されています）

３．Foundation 層の機能

Foundation 層では将棋プログラムを実現するためのプリミティブな機能を提供します．

(1) Atomic

アトミック操作を行うための変数と操作のクラスです．GCC のビルトイン関数を使用して実装しています．インクリメント，デクリメント，テストアンドスワップなどの機能を提供します．

並列性を高めるためには共有リソースのロック粒度をできるだけ小さくする必要があります．アトミック操作で対応可能な処理はそれを使用することによりロックを回避することができます．

(2) SpinLock

相互排他機能を提供するクラスです．glibc の NPTL スピンロック関数を使用しています．スピンロックは OS のスケジューラに依存しないため軽量です．インテルプロセッサの場合，物理コア数だけスレッドを生成し，それぞれがイベント待ちなどで眠ることなく走り続けた場合に良い並列性能を期待することができます．

(3) Semaphore
スレッド間で同期をとるためのクラスです. glibc の POSIX セマフォを使用しています.

(4) Thread
glibc の NPTL を使用したスレッドクラスです. スレッドの生成, スレッドの終了確認, スレッド合流, 結果の保持, リソースのクリーンアップ機能を提供します.

４. Game 層の機能

合法手生成機能は Bitboard により実装を行いました. libshogi はルール通り将棋を指す最小限の機能のみを提供します.

(1) Shogi
合法手の生成, 着手, 棋譜の読み込みなどの将棋に特化した機能を提供するクラス群です.

(2) Communication
外部プログラム, フロントエンド GUI との通信を行うための機能を提供するクラスです.

５. おわりに

libshogi が目標とするのは第一に分かり易いということです. 将棋プログラムに興味を持った学生が, すぐに Strategy 層から開発を開始する助けになればというのが作者らの願いです.

参考文献

1) 山下 宏: "YSS -『コンピュータ将棋の進歩 2 』以降の改良点", 『アマトップクラスに迫る - コンピュータ将棋の進歩 5』, 1 章, 共立出版, 2005.

2) 池泰弘: 『コンピュータ将棋のアルゴリズム』, 工学社, 2005.

3) 保木邦仁: "Bonanza 4.1.3 ソースコード", 『プロ棋士に並ぶ - コンピュータ将棋の進歩 6』, 1 章, 共立出版, 2012.

4) 山本一成, 竹内聖吾, 金子知適, 田中哲郎: "コンピュータ将棋における Magic Bitboards の提案と実装", 第15回ゲームプログラミングワークショップ, pp42-48, 2010

5) 平岡拓也: "Apery ソースコード", https://github.com/HiraokaTakuya/apery , commit at Tue Nov 24 20:54:35 2015

6) 磯崎元洋: "やねうら王mini ソースコード", https://github.com/yaneurao/YaneuraOu commit at Mon Mar 7 11:31:40 2016