

将棋プログラム用ライブラリ libshogi

藤井宏行[†] 荻田稔真[†] 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。文法のみで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並列探索を行う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。

現在は、基本的なデータ構造の実装が完了し、ルール通りに指すための将棋機能について実装を進めています。

このライブラリには libshogi という名前を付けました。作者らが十分な品質に至ったと判断した時点でソースコードを公開する予定です。

2. 概要

図-1 に将棋プログラムの階層モデルを示します。これは、将棋プログラムがこのような階層構造を持つべきであるという意味ではなく、このような実装も可能であろうという例です。

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。

Foundation 層では、アトミック操作、ロック、スレッド、リスト、二分木、ハッシュ木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。

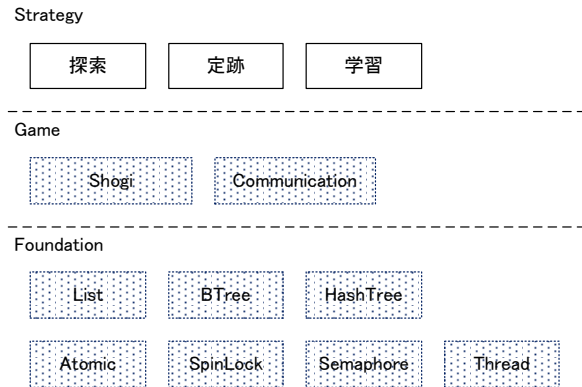


図-1 将棋プログラムの階層モデル

libshogi は Boost や STL のコンテナクラス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。昨年からの改良点としてハッシュ木を平衡木（赤黒木）に変えました。ハッシュ値のランダム性から木が大きく偏ることはないと考えていましたが、単純な二分木だと手が広い局面でメモリ使用量が 10GB を超えることもあり、実用的ではないことが分かりました。

(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。

(5) List

連結リストのコンテナクラスです。要素の追加、削除、イテレータ機能を提供します。

(6) BTree

二分木のコンテナクラスです。ノードの追加、探索機能を提供します。

(7) HashTree

局面のハッシュ値をキーとして局面情報を検索するための二分木コンテナクラスです（いわゆるハッシュ木とは少し違います）キー値によるノードの追加、探索、値の更新機能を提供します。

4. Game 層の機能

合法手生成などの機能に池泰弘さんが開発された「れさびょん」を使用させていただく予定です。

(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. 将棋プログラムの特徴

libshogi は学習用を想定した、将棋プログラムを作るためのライブラリです。同時に実用的であることを確認するために、将棋プログラムの開発も行っています。この将棋プログラムの特徴を紹介します。

(1) 評価関数

評価関数は Bonanza 型の KKP, KPP を採用しています。学習方法は Bonanza と異なっており、パーセプトロンとオンライン学習を試しています。この方法が優れているところは、少ない教師例と学習である程度の一致率を達成できることです。

(2) 探索

$\alpha\beta$ 法に加えて Null Move Pruning などの枝刈り手法を試しています。また、スレッドによる並列探索に加え、疎結合クラスタによる探索の検証も行っています。

6. 今後の目標

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。

一方で実用性についても磨きをかけていければと考えています。libshogi を土台として少しでも強い将棋プログラムが開発できるよう改良を続けていければと思います。

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。