

6J-06

LablGtk3 と cairo2 を用いた OCaml による GTK3 ベース GUI アプリケーションとしての証明木図示ツール Traf の実装

浅野貴信† 川端英之‡ 弘中哲夫‡

広島市立大学情報科学部情報工学科† 広島市立大学大学院情報科学研究科‡

1 はじめに

プログラミング言語 OCaml を用いて GTK ベースでの GUI アプリケーションを作成するには、ライブラリ LablGtk が利用されている。最新の LablGtk3 は、GTK3 と Cairo に対応されて間もない点も然ることながら、GTK2 と 3 の関係と同様に、LablGtk2 との互換性が低く移行が難しいため、実装例はあまり見られない。そこで我々は、LablGtk2 から 3 への移行に関する知見の蓄積・共有を目的とし、証明木図示ツール Traf の LablGtk3 と cairo2 への移行を試みた。本発表では、この取り組みを通して得られた、プログラムの LablGtk2 から 3 への移行に関する知見について報告する。

2 GTK と LablGtk

2.1 概要

GTK (GIMP Tool Kit) はオープンソースの GUI ウィジェットツールキットの一つで、様々な環境で広く利用されている [1]。GTK は C 言語で記述された柔軟性の高いライブラリで、C++ や Python など様々なプログラミング言語で利用するためのバインディングが開発されている。GTK を OCaml から利用するためのバインディングは LablGtk と呼ばれる [2]。

2.2 GTK のバージョンの変遷と現状

GTK2 は 2002 年に公開された。その後 2011 年に GTK3 がリリースされているが、長期にわたりいずれのバージョンも利用可能であった。しかしながら、2020 年 12 月 16 日に GTK 4.0 が公開されたタイミングで、GTK2 は *End of Life* となった。GTK ユーザには、GTK3 あるいは GTK4 への早急な移行が求められている。

GTK2 から GTK3 への移行にあたっては、GTK が依存しているライブラリに大きな変更があった [3]。例えば、GTK が用いる GDK の描画プリミティブに密接に関連する GdkDrawable 型のデータ構造が利用不可となった。結果として、描画機能やイメージデータを使用するアプリケーションは、ベクターグラフィックスライブラリ Cairo [4] を用いるように変更を強いられることになった。

GTK の OCaml バインディングである LablGtk は、GTK3 に対応した LablGtk3 が提供されているものの、

GdkDrawable の欠落分を補うのに便利な GdkWindow のための一部の関数（例えば `gdk_window_get_width`）を呼ぶ仕組みが無い¹など、LablGtk2 ベースのアプリケーションの移行には少なからぬ手数が掛かる状況である。描画機能の使用にあたっては、LablGtk3 で削除された GDraw モジュールの drawable クラスの代わりに、Cairo の OCaml バインディングである cairo2 ライブラリを用いることになる。

3 証明木図示ツール Traf

証明木図示ツール Traf [5] は、定理証明支援系 Coq [6] を Proof General [7] 経由で対話的に用いる際に証明の構造を随時グラフィカルに表示できる GUI アプリケーションで、LablGtk2 を用いて OCaml で記述されている [8]。これまで、Traf を C++ で書き換えて GTK3 ベースにした上で機能を強化するなどの試みは続けられているが [9]、未だオープンソース化には至っておらず、OCaml 版 Traf の GTK3 対応が早急に求められている。

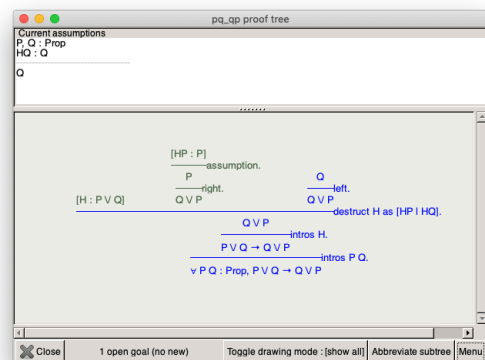


図 1: LablGtk2 版 Traf

図 1 に Traf のスクリーンショットを示す。図には命題 $\forall P Q, P \wedge Q \rightarrow Q \wedge P$ を証明する途中の様子が描かれている。図からわかる通り、Traf の実装には図形の描画機能が不可欠であるが、LablGtk2 と LablGtk3 の互換性は低く、描画のための一連の記述を置き換える必要がある。

4 GTK3 ベース Traf の開発

我々は、LablGtk2 ベースの Traf に対して、LablGtk3 および cairo2 ライブラリを用いた GTK3 ベース記述への書き換えを行うことを通して、LablGtk2 アプリケーションを LablGtk3 ベースに移行するために必要な知見の整理を試みた。以下では、移行のために行った記述変更の内容を具体的に述べる。

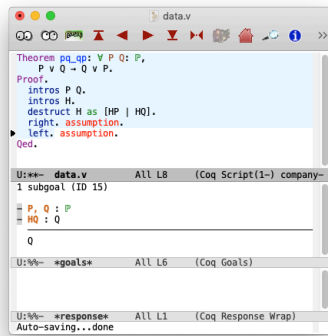
Implementing a proof tree viewer Traf based on LablGtk3 and cairo2 as a GTK3-based GUI application in OCaml

Takanobu Asano† Hideyuki Kawabata‡ Tetsuo Hironaka‡

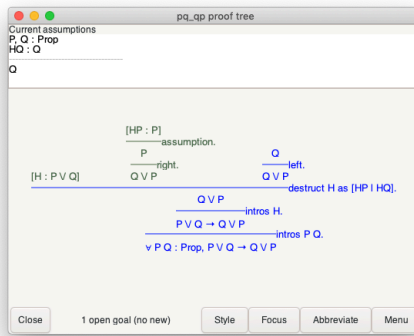
†Department of Computer and Network Engineering, Hiroshima City University

‡Graduate School of Information Sciences, Hiroshima City University

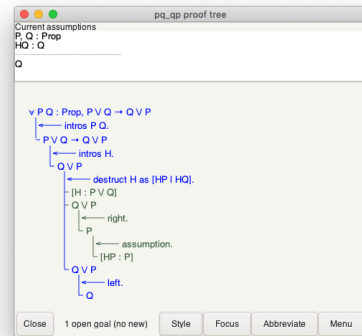
¹この関数も GTK 4.0 では削除された。



(a) Proof General を使う様子



(b) Traf による証明図描画



(c) Traf による証明図描画 (Fitch 風)

図 2: LablGtk3 版の Traf を使用する様子

4.1 大幅な変更が必要であったもの

4.1.1 GDraw.drawable

LablGtk3 では GDraw モジュールの drawable クラスが廃止されている。また, drawable クラスの描画メソッドの実装に用いられていた (間接的に `gdk_draw_line` を呼ぶための `Draw.line` などの) Gdk.Draw モジュールの関数も廃止されている。これらは全て, Cairo ライブラリを用いた実装が必要となる。Traf では GDraw.drawable のサブクラスを生成して描画に用いていたため, 元の GDraw.drawable を模したクラスを用意して対処した。例えば長方形を描画するメソッドは次のようにした:

```
method rectangle ~x ~y ~width ~height ?(filled=false) () =
  let cr = self#get_cr in
  Cairo.set_line_width cr (float line_width);
  Cairo.rectangle cr (float x) (float y)
    ~w:(float width) ~h:(float height);
  if filled then Cairo.fill cr else Cairo.stroke cr
```

なお, 上のコード中の `cr` は `Cairo.context` 型の値で, GTK2 における Graphics Contexts の代替に当たる。

4.1.2 Gdk.GC

GTK2 の描画要素の属性値設定関数 (例えば `gdk_gc_set_line_attributes`) が利用不能となり, LablGtk2 でも Gdk.GC モジュールが削除されたため, 代替の関数が必要となった。上のコード例のように Cairo モジュールの `set_line_width` などの関数を用い, `Cairo.context` に対して属性を設定するようにした。

4.2 微修正が必要であったもの

- GMisc モジュールの関数 `drawing_area` の仕様や, `drawing_area` クラスのオブジェクトへのコールバック関数の設定手順に変更があった。
- GPango モジュールの `context_rw` クラスが `context` クラスにマージされたことに対処した。
- GPango モジュールの `font_description` 関数が `font_description_from_string` に変更された。
- GPango.context クラスの `create_layout` メソッドの仕様が変更, Pango.Layout モジュールの `create` 関数の戻り値のオブジェクトが LablGtk3 では GPango

モジュールの layout クラスに内包されて扱われるようになった。移植に際し, `create_layout` メソッド呼び出し後に適宜 `as_layout` メソッドにより内容を取り出すようにした。

- GWindow モジュールの `about_dialog` 関数で得られるオブジェクトの “response” に対するコールバック関数が実際に呼ばれたとき (Close ボタン押下時) に引数として渡される値に違いがあった。
- GWindow モジュールの `dialog` 関数の名前付き引数 `no_separator` が削除されたことに対処した。
- プログラムの実行開始時に `GMain.init()` を呼び出すようにした (LablGtk2 では必須ではなかった)。

以上のほか, 環境設定のためにいくつかのファイル (`configure` や `Makefile.in`) を書き換えて, LablGtk3 および Cairo を参照するように変更した。

図 2 に, LablGtk3 版の Traf の動作の様子を示す。

5 まとめ

LablGtk2 ベースのアプリケーションの早急な移行には, GDraw.drawable の代替物を Cairo で自作することが効果的と思われる。また, 同等物を LablGtk3 レベルで提供できればユーザにとっての利便性は高いだろう。

より網羅的な情報の整理が今後の課題である。

参考文献

- [1] GTK+ 3 Reference Manual. <https://developer.gnome.org/gtk3/stable/>
- [2] LablGtk. <https://github.com/garrigue/lablgtk>
- [3] The GNOME Project: Migrating from GTK+ 2.x to GTK+ 3. <https://developer.gnome.org/gtk3/stable/gtk-migrating-2-to-3.html>
- [4] Cairo graphics library. <https://www.cairographics.org>
- [5] H. Kawabata, Y. Tanaka, M. Kimura, T. Hironaka: Traf: A Graphical Proof Tree Viewer Cooperating with Coq Through Proof General, APLAS 2018, LNCS 11275, pp.157–165, 2018.
- [6] The Coq Proof Assistant. <https://coq.inria.fr/>
- [7] Proof General. <http://proofgeneral.github.io/>
- [8] Traf. <https://github.com/hide-kawabata/traf>
- [9] 古谷, 川端, 弘中, 定理証明支援系 Coq と連携した証明木図示ツールにおける大域的および局所的な情報把握支援機能の改善, 第 82 回情報処全大, 4K-01, pp.1-173–1-174, Mar. 2020.