関数リアクティブプログラミング言語による小規模組込みシステムのプログラミング

渡部卓雄（東京工業大学 情報理工学院）

小規模組込みシステム向け関数リアクティブプログラミング(FRP)言語Emfrpは、8bitマイクロコントローラでも動作する純粋関数型言語である。本言語では、センサーの計測値などの時間とともに変化するデータを時変値と呼ばれるオブジェクトとして抽象化し、それらの関係を副作用のない関数として定義することで組込みシステムの動作を記述する。本発表では、Emfrpとそのプログラム例に加え、現在進行中の関連プロジェクトについて紹介する。

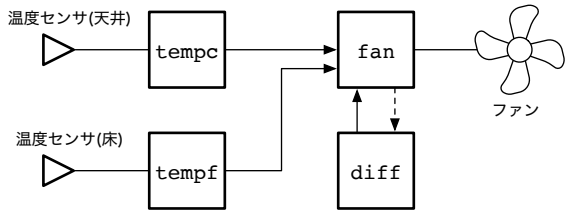
FRP言語Emfrp

関数リアクティブプログラミング(FRP)は，時間と共に変化する値を抽象化した時変値(time-varying value)を用いることで，リアクティブシステムの宣言的な記述を支援するプログラミングパラダイムである．FRPにより，コールバック，イベントループ，ポーリング等を排し，組込みシステムの動作を見通しよく記述することが可能になる．

```
module FanController       # module name
in  tempc, tempf : Float, # temperature sensors
out fan          : Bool   # fan switch
use Std                   # standard library

node fan = abs(tempc - tempf) >= diff
node diff = 5.0 + if fan@last then -0.5 else 0.5
```

Emfrpによるファン制御プログラム



時変値間の依存関係

我々はマイクロコントローラなどの小規模システムを対象とする純粋FRP言語Emfrp[1]を開発した．右上は簡単なEmfrpプログラムの例で，天井と床に設置した温度センサーで計測した値の差が規定値以上のときにファンをONにし，室内の空気を攪拌して温度差を小さくする．このプログラムでは，計測値の差が規定値付近を動くときにファンのON/OFFが頻繁に行われないよう，簡単なヒステリシス制御を行なっている．プログラム中の tempc，tempf，fan，diff はすべて時変値であり，これらの間には右下のグラフで示す依存関係がある．Emfrpプログラムの実行は，この依存関係に沿って時変値の値を更新し続けることに相当する．またEmfrpでは，プログラムが使用する（スタックを含む）メモリの大きさがコンパイル時に決定されることと，時変値の1回の更新（プログラム中のnodeで始まる文における=の右辺の計算）は必ず停止することが保証されている．したがって，Emfrpはマイクロコントローラのような限られたリソースで動作するプログラムの開発に適している．

Emfrpコンパイラはソースコードを受け取ってCのコードを生成する．生成されたコードは時変値の更新を続ける部分であり，プラットフォーム非依存である．一方入出力を行う部分はプラットフォーム依存であり，別途C等で記述する必要がある．現在リリース済みのEmfrpコンパイラはRubyで記述されており，gem install emfrp でインストールできる（Ruby 2.7.xが必要）．

[1] Sawada, K. & T. Watanabe, "Emfrp: A Functional Reactive Programming Language for Small-Scale Embedded Systems", CROW 2016, ACM, 2016.

現在進行中の関連プロジェクト

Emfrp^{BCT}　上で述べたように，Emfrpではプログラムが使用するメモリサイズがコンパイル時に決定でき，かつ時変値の更新がスタックすることはない．これらは小規模組込みシステムにおいて有用な性質であるが，実際にはデータ型や関数の再帰的な定義や高階関数の利用を禁止することによって達成されている．リストや木などの再帰的データやそれらに伴う再帰的な関数は組込みシステムにおいても有用であるため，我々はデータサイズを陽に指定することのできる型システムBCT(Bounded Construction Types)を定義した．Emfrpにこの型システムを導入した言語Emfrp^{BCT}を設計してその意味論を与え，いくつかの性質を証明した[2].

[2] Yokoyama, A., S. Moriguchi & T. Watanabe, "A Functional Reactive Programming Language for Small-Scale Embedded Systems with Recursive Data Types", J. of Information Processing, IPSJ, 2021 (to appear).

XStorm　多くの組込みシステムは状態を持ち，その設計にはStateChart等の状態遷移図が用いられる．Emfrpのような純粋なFRP言語においても状態依存の動作は表現できるが，その記述は煩雑になりがちになる．XStormはEmfrpの後継言語XFRPに状態依存動作を記述するための機構（switch拡張）を導入したものであり，これによって状態遷移図をベースとした設計をベースにFRPにもとづくプログラムを簡潔に記述することが可能になることを示した[3].

[3] 松村有倫, 渡部卓雄, 組込みシステム向けFRP言語における状態依存動作のための抽象化機構, 情報処理学会論文誌（プログラミング）, 13(2), pp. 1-13, 2020.

並列実行　上で述べたように，FRPにもとづくプログラムの実行は依存関係に沿って時変値を更新し続けることとみなすことができる．一般に時変値間の依存関係はDAGを構成するため，いくつかの時変値の更新は並列に行うことが可能である．最近はマイクロコントローラにおいてもマルチコア化が進んでいるため，我々はEmfrpの後継言語であるXFRPの並列実行モデルXFRP-Paraを提案し，プロトタイプ処理系の実装と評価を通してその有効性を示した[4]．加えて，GPGPU上での並列実行方式の提案と評価を行なった[5].

[4] Sakurai, Y. & T. Watanabe, Towards a Statically Scheduled Parallel Execution of an FRP Langauge for Embedded Systems, REBLS '19, ACM, 2019.
[5] Sakurai, Y., S. Moriguchi & T. Watanabe, Functional Reactive Programming for Embedded Systems with GPGPUs, ICSCA '21, ACM, 2021.