

Coqを使ったデジタルデータ放送におけるストリーム データ処理の形式化と検証

今井, 宣洋
有限会社ITプランニング

<https://hdl.handle.net/2324/1520942>

出版情報 : MI lecture note series. 61, pp.53-58, 2015-03-06. 九州大学マス・フォア・インダストリ
研究所
バージョン :
権利関係 :



Coq を使った

デジタルデータ放送における
ストリームデータ処理の形式化と検証

**Formalization and verification of
stream data processing for
datacasting in Coq**

今井宜洋 Yoshihiro Imai

有限会社 IT プランニング

今井宜洋 Yoshihiro Imai

Coq を使った デジタルデータ放送における ストリームデータ処理の形式化と検証 Formalization and verification of stream data processing

今井宜洋 Yoshihiro Imai

Coq を使った デジタルデータ放送における ストリームデータ処理の形式化と検証 Formalization and verification of stream data processing

発表すること

- ▶ (有) IT プランニング について
- ▶ Coq を使用した案件例 (データ放送)
- ▶ 我々が Coq を使う理由

有限会社 IT プランニング

IT プランニングについて

IT Planning, Inc.

- ▶ 有限会社 IT プランニング
- ▶ 3 人のエンジニアと 5 人のアルバイト
- ▶ <http://www.itpl.co.jp/>

今井宜洋 Yoshihiro Imai

Coq を使った デジタルデータ放送における ストリームデータ処理の形式化と検証 Formalization and verification of stream data processing

今井宜洋 Yoshihiro Imai

Coq を使った デジタルデータ放送における ストリームデータ処理の形式化と検証 Formalization and verification of stream data processing

IT プランニングについて

- ▶ 独立系ソフトウェアベンダ
- ▶ 仕様言語: OCaml, Scala, Haskell, F#, Java, etc.

有限会社 IT プランニング

ITプランニングについて

顧客



今井宣洋 Yoshihiro Imai

有限会社 IT プランニング

Coq を使った デジタルデータ放送における ストリームデータ処理の形式化と検証 Formalization and verification of stream data processing

地上デジタル放送

2012年、日本では地上デジタル放送へ完全移行した。



今井宣洋 Yoshihiro Imai

有限会社 IT プランニング

Coq を使った デジタルデータ放送における ストリームデータ処理の形式化と検証 Formalization and verification of stream data processing

地上デジタル放送

緊急津波速報



今井宣洋 Yoshihiro Imai

有限会社 IT プランニング

Coq を使った デジタルデータ放送における ストリームデータ処理の形式化と検証 Formalization and verification of stream data processing

地上デジタル放送

緊急地震速報



今井宣洋 Yoshihiro Imai

有限会社 IT プランニング

Coq を使った デジタルデータ放送における ストリームデータ処理の形式化と検証 Formalization and verification of stream data processing

地上デジタル放送

開票速報



今井寛洋 Yoshihiro Imai

有限会社 IT プランニング

Coq を使った デジタルデータ放送における ストリームデータ処理の形式化と検証 Formalization and verification of stream data processing

地上デジタル放送

マラソン

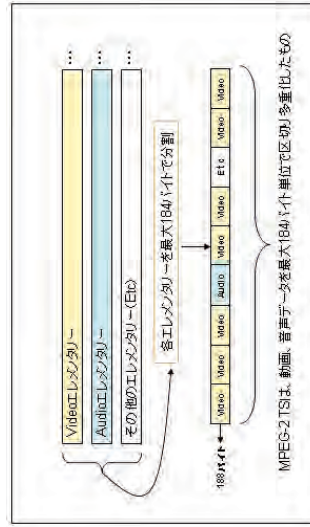


今井寛洋 Yoshihiro Imai

有限会社 IT プランニング

Coq を使った デジタルデータ放送における ストリームデータ処理の形式化と検証 Formalization and verification of stream data processing

Mpeg2-TS



今井寛洋 Yoshihiro Imai

有限会社 IT プランニング

Coq を使った デジタルデータ放送における ストリームデータ処理の形式化と検証 Formalization and verification of stream data processing

受注案件

- ▶ 放送データを解析しデジタルデータコンテンツを取り出す
 - ▶ データコンテンツを検証
- 要求
- ▶ 高い処理スピード
 - ▶ データの正確性
- OCaml で開発した

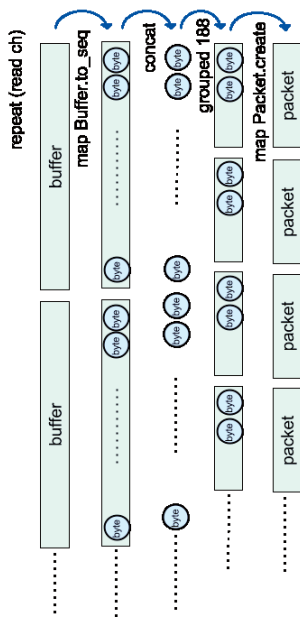
今井寛洋 Yoshihiro Imai

有限会社 IT プランニング

Coq を使った デジタルデータ放送における ストリームデータ処理の形式化と検証 Formalization and verification of stream data processing

take_packets

take_packets : seq bytes → seq bytes



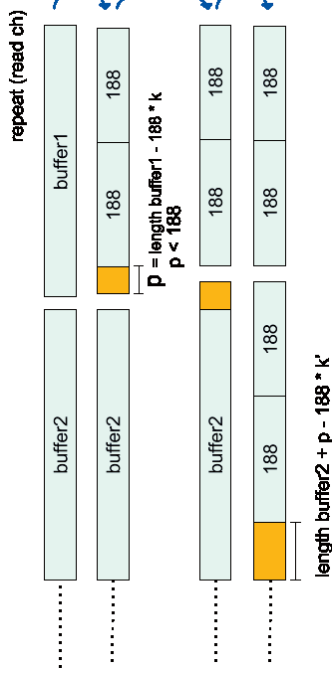
CoInductive seq (A: Type) : Type :=

- | SNil : seq A
- | SCons : A → seq A → seq A.

Variables bytes : Set.

Variables sub : bytes → nat → nat → nat → bytes.

take_packets'



証明した性質 (の一部)

Goal forall bufs : seq bytes ,
take_packets bufs == take_packets' bufs .

Goal forall buffers : seq bytes ,
ForAll
(fun packet => length packet = 188)
(take_packets' buffers) .

Goal forall bufs : seq bytes ,
concat (take_packets' bufs) ==
concat bufs .

証明

- ▶ 定義をあわせて数 100 行程度
- ▶ テストよりも楽だったかも

今井寛洋 Yoshihiro Imai

有限会社 IT プランニング

Coq を使った デジタルデータ放送における ストリームデータ処理の形式化と検証 Formalization and verification of stream data processing

良かったこと

- ▶ 証明したことについては疑う必要がない前提でより複雑な部分の開発に望めた
- ▶ 全体のうち関数一つだけを抜き出して検証ができた

今井寛洋 Yoshihiro Imai

有限会社 IT プランニング

Coq を使った デジタルデータ放送における ストリームデータ処理の形式化と検証 Formalization and verification of stream data processing

Coq を使う理由

我々の考え方

- ▶ 信頼 → 営業活動
 - ▶ ソフトウェアの品質 → 信頼
- ここでの品質とは、お客さんの前で致命的なエラーが起きない、実行時エラーが起きないということ

今井寛洋 Yoshihiro Imai

有限会社 IT プランニング

Coq を使った デジタルデータ放送における ストリームデータ処理の形式化と検証 Formalization and verification of stream data processing

Coq を使う理由

品質のために我々は静的型付き関数型言語を使っている

- ▶ null がない
- ▶ 型安全
- ▶ ユニットテストが書きやすい

今井寛洋 Yoshihiro Imai

有限会社 IT プランニング

Coq を使った デジタルデータ放送における ストリームデータ処理の形式化と検証 Formalization and verification of stream data processing

Coqの良い所

- ▶ 強力な仕様記述能力
- ▶ 証明済みのプログラムが得られる (Extraction)
- ▶ 自動証明 (tactic)

まとめ

- ▶ 民間企業では信頼関係が仕事を生む
- ▶ 納品物の品質が信頼関係を構築する
- ▶ Coqがもたらす品質はテストでは得られない

証明に向いている検証項目

- "forall x: A, ..."
- ▶ "A"はリスト型、関数型など無限の要素を持つ型
- 典型的な型
- ▶ forall x, f x = g x
 - ▶ forall x, g (f x) = x
 - ▶ forall x, P x → P (f x)