One Part of Shannon's Source Coding Theorem

Quentin Hibon

April 20, 2020

Abstract

This document contains a proof of the necessary condition on the code rate of a source code, namely that this code rate is bounded by the entropy of the source. This represents one half of Shannon's source coding theorem, which is itself an equivalence.

This proof is taken directly from the textbook [1], and transcribed rather literally into Isabelle. It is thus easier to keep the textbook proof in mind to understand this formal proof.

Contents

theory *Source-Coding-Theorem*
imports *HOL−Probability.Information*
begin

1 Basic types

type-synonym *bit* = *bool*
type-synonym *bword* = *bit list*
type-synonym *letter* = *nat*

type-synonym *'b word* = *'b list*

type-synonym *'b encoder* = *'b word* ⇒ *bword*
type-synonym *'b decoder* = *bword* ⇒ *'b word option*

2 Locale for the source coding theorem

locale *source-code* = *information-space* +
  fixes *fi* :: *'b* ⇒ *real*
  fixes *X* :: *'a* ⇒ *'b*

  assumes *distr-i*: *simple-distributed M X fi*
  assumes *b-val*: *b* = *2*

  fixes *enc*::*'b encoder*
  fixes *dec*::*'b decoder*
  assumes *real-code*:
  dec (*enc x*) = *Some x*
  enc w = [] ⟷ *w* = []
  x ≠ [] ⟶ *enc x* = *enc* [*hd x*] @ *enc* (*tl x*)

3 Source coding theorem, direct: the entropy is a lower bound of the code rate

context *source-code*
begin

3.1 The letter set

definition *L* :: *'b set* **where**
  L ≡ *X* ‘ *space M*

lemma *fin-L*: *finite L*
  ⟨*proof*⟩

lemma *emp-L*: *L* ≠ {}
  ⟨*proof*⟩

3.2 Codes and words

abbreviation *real-word* :: *'b word* ⇒ *bool* **where**
  real-word w ≡ (*set w* ⊆ *L*)

abbreviation *k-words* :: *nat* ⇒ (*'b word*) *set* **where**
  k-words k ≡ {*w*. *length w* = *k* ∧ *real-word w*}

lemma *rw-tail*:
  assumes *real-word w*

shows $w = [] \lor$ *real-word* $(tl\ w)$
⟨*proof*⟩

definition *code-word-length* :: $'e$ *encoder* $\Rightarrow\ 'e \Rightarrow$ *nat* **where**
code-word-length e l = *length* (*e* [*l*])

abbreviation *cw-len* :: $'b \Rightarrow$ *nat* **where**
cw-len l $\equiv$ *code-word-length enc l*

definition *code-rate* :: $'e$ *encoder* $\Rightarrow ('a \Rightarrow 'e) \Rightarrow$ *real* **where**
code-rate e Xo = *expectation* ($\lambda a.$ (*code-word-length e* ((*Xo*) *a*)))

lemma *fi-pos*: $i\in L \Longrightarrow 0 \leq fi\ i$
⟨*proof*⟩

lemma (**in** *prob-space*) *simp-exp-composed*:
assumes *X*: *simple-distributed M X Px*
shows *expectation* ($\lambda a.\ f\ (X\ a)$) = ($\sum x \in X$‘*space M*. $f\ x * Px\ x$)
⟨*proof*⟩

lemma *cr-rw*:
code-rate enc X = ($\sum i \in X$ ‘ *space M*. *fi i* $*$ *cw-len i*)
⟨*proof*⟩

abbreviation *cw-len-concat* :: $'b$ *word* $\Rightarrow$ *nat* **where**
cw-len-concat w $\equiv$ *foldr* ($\lambda x\ s.$ (*cw-len x*) + *s*) *w 0*

lemma *cw-len-length*: *cw-len-concat w* = *length* (*enc w*)
⟨*proof*⟩

lemma *maj-fold*:
assumes $\bigwedge l.\ l\in L \Longrightarrow f\ l \leq$ *bound*
assumes *real-word w*
shows *foldr* ($\lambda x\ s.\ f\ x + s$) *w 0* $\leq$ *length w* $*$ *bound*
⟨*proof*⟩

definition *max-len* :: *nat* **where**
max-len = *Max* (($\lambda x.$ *cw-len x*) ‘ *L*)

lemma *max-cw*:
$l \in L \Longrightarrow$ *cw-len l* $\leq$ *max-len*
⟨*proof*⟩

3.3 Related to the Kraft theorem

definition $\mathcal{K}$ :: *real* **where**
$\mathcal{K}$ = ($\sum i\in L.$ *1* / *b* ^ (*cw-len i*))

lemma *pos-cw-len*: *0* < *1* / *b* ^ *cw-len i* ⟨*proof*⟩

lemma $\mathcal{K}$-*pos*: $0 < \mathcal{K}$
⟨*proof*⟩

lemma $\mathcal{K}$-*pow*: $\mathcal{K} = (\sum i{\in}L.\ 1\ /\ b\ powr\ cw\text{-}len\ i)$
⟨*proof*⟩

lemma *k-words-rel*:
$k\text{-}words\ (Suc\ k) = \{w.\ (hd\ w \in L \land tl\ w \in k\text{-}words\ k \land w \neq [])\}$
⟨*proof*⟩

lemma *bij-k-words*:
shows $bij\text{-}betw\ (\lambda wi.\ Cons\ (fst\ wi)\ (snd\ wi))\ (L \times k\text{-}words\ k)\ (k\text{-}words\ (Suc\ k))$
⟨*proof*⟩

lemma *finite-k-words*: $finite\ (k\text{-}words\ k)$
⟨*proof*⟩

lemma *cartesian-product*:
fixes $f{::}('c \Rightarrow real)$
fixes $g{::}('d \Rightarrow real)$
assumes *finite A*
assumes *finite B*
shows $(\sum b{\in}B.\ g\ b) * (\sum a{\in}A.\ f\ a) = (\sum ab{\in}A{\times}B.\ f\ (fst\ ab) * g\ (snd\ ab))$
⟨*proof*⟩

lemma $\mathcal{K}$-*power*:
shows $\mathcal{K}\,\hat{}\,k = (\sum w \in (k\text{-}words\ k).\ 1\ /\ b\,\hat{}(cw\text{-}len\text{-}concat\ w))$
⟨*proof*⟩

lemma *bound-len-concat*:
shows $w \in k\text{-}words\ k \Longrightarrow cw\text{-}len\text{-}concat\ w \leq k * max\text{-}len$
⟨*proof*⟩

3.4 Inequality of the kraft sum (source coding theorem, direct)

3.4.1 Sum manipulation lemmas and McMillan theorem

lemma *sum-vimage-proof*:
fixes $g{::}nat \Rightarrow real$
assumes $\bigwedge w.\ f\ w < bd$
shows $finite\ S \Longrightarrow (\sum w{\in}S.\ g\ (f\ w)) = (\sum\ m{=}0..{<}bd.\ (card\ ((f-`\{m\}) \cap S)$
$)* g\ m)$
(**is** $\text{-} \Longrightarrow \text{-} = (\sum\ m{=}0..{<}bd.\ ?ff\ m\ S))$
⟨*proof*⟩

lemma *sum-vimage*:
fixes $g{::}nat \Rightarrow real$
assumes *bounded*: $\bigwedge w.\ w \in S \Longrightarrow f\ w < bd$ **and** $0 < bd$

assumes *finite*: *finite S*
shows $(\sum w{\in}S.\ g\ (f\ w)) = (\sum\ m{=}0..{<}bd.\ (card\ ((f{-}`\{m\}) \cap S)\ ) * g\ m)$
(**is** *?s1 = ?s2*)
⟨*proof*⟩

lemma $\mathcal{K}$*-rw*:
$(\sum w \in (k\text{-}words\ k).\ 1\ /\ b\hat{\ }(cw\text{-}len\text{-}concat\ w)) = (\sum m{=}0..{<}Suc\ (k{*}max\text{-}len).$
$card\ (k\text{-}words\ k\ \cap$
$((cw\text{-}len\text{-}concat)\ -`\ \{m\})) * (1\ /\ b\hat{\ }m))$ (**is** *?L = ?R*)
⟨*proof*⟩

definition *set-of-k-words-length-m* :: *nat* ⇒ *nat* ⇒ *'b word set* **where**
$set\text{-}of\text{-}k\text{-}words\text{-}length\text{-}m\ k\ m = \{xk.\ xk \in k\text{-}words\ k\} \cap (cw\text{-}len\text{-}concat){-}`\{m\}$

lemma *am-inj-code*: *inj-on enc* $((cw\text{-}len\text{-}concat){-}`\{m\})$ (**is** *inj-on - ?s*)
⟨*proof*⟩

lemma *img-inc*: $enc`cw\text{-}len\text{-}concat{-}`\{m\} \subseteq \{bl.\ length\ bl = m\}$ ⟨*proof*⟩

lemma *bool-lists-card*: $card\ \{bl{::}bool\ list.\ length\ bl = m\} = b\hat{\ }m$
⟨*proof*⟩

lemma *bool-list-fin*: $finite\ \{bl{::}bool\ list.\ length\ bl = m\}$
⟨*proof*⟩

lemma *set-of-k-words-bound*:
shows $card\ (set\text{-}of\text{-}k\text{-}words\text{-}length\text{-}m\ k\ m) \leq b\hat{\ }m$ (**is** *?c ≤ ?b*)
⟨*proof*⟩

lemma *empty-set-k-words*:
assumes $0 < k$
shows $set\text{-}of\text{-}k\text{-}words\text{-}length\text{-}m\ k\ 0 = \{\}$
⟨*proof*⟩

lemma $\mathcal{K}$*-rw2*:
assumes $0 < k$
shows $(\sum m{=}0..{<}Suc\ (k * max\text{-}len).\ card\ (set\text{-}of\text{-}k\text{-}words\text{-}length\text{-}m\ k\ m)/\ b\hat{\ }m)$
$\leq (k * max\text{-}len)$
⟨*proof*⟩

lemma $\mathcal{K}$*-power-bound* :
assumes $0 < k$
shows $\mathcal{K}\hat{\ }k \leq k * max\text{-}len$
⟨*proof*⟩

theorem *McMillan* :
shows $\mathcal{K} \leq 1$
⟨*proof*⟩

lemma *entropy-rw*: $\mathcal{H}(X) = -(\sum i \in L.\ fi\ i * log\ b\ (fi\ i))$
  ⟨*proof*⟩

3.4.2 Technical lemmas about the logarithm

lemma *log-mult-ext3*:
 $0 \leq x \Longrightarrow 0 < y \Longrightarrow 0 < z \Longrightarrow x * log\ b\ (x{*}y{*}z) = x * log\ b\ (x{*}y) + x * log\ b\ z$
  ⟨*proof*⟩

lemma *log-mult-ext2*:
 $0 \leq x \Longrightarrow 0 < y \Longrightarrow x * log\ b\ (x{*}y) = x * log\ b\ x + x * log\ b\ y$
  ⟨*proof*⟩

3.4.3 KL divergence and properties

definition *KL-div* $::'b\ set \Rightarrow ('b \Rightarrow real) \Rightarrow ('b \Rightarrow real) \Rightarrow real$ **where**
 $KL\text{-}div\ S\ a\ d = (\sum\ i \in S.\ a\ i * log\ b\ (a\ i\ /\ d\ i))$

lemma *KL-div-mul*:
 assumes $0 < d\ d \leq 1$
 assumes $\bigwedge i.\ i{\in}S \Longrightarrow 0 \leq a\ i$
 assumes $\bigwedge i.\ i{\in}S \Longrightarrow 0 < e\ i$
shows $KL\text{-}div\ S\ a\ e \geq KL\text{-}div\ S\ a\ (\lambda i.\ e\ i\ /\ d)$
  ⟨*proof*⟩

lemma *KL-div-pos*:
 fixes $a\ e{::}'b \Rightarrow real$
 assumes *fin*: *finite S*
 assumes *nemp*: $S \neq \{\}$
 assumes *non-null*: $\bigwedge i.\ i{\in}S \Longrightarrow 0 < a\ i\ \bigwedge i.\ i{\in}\ S \Longrightarrow 0 < e\ i$
 assumes *sum-a-one*: $(\sum\ i \in S.\ a\ i) = 1$
 assumes *sum-c-one*: $(\sum\ i \in S.\ e\ i) = 1$
shows $0 \leq KL\text{-}div\ S\ a\ e$
  ⟨*proof*⟩

lemma *KL-div-pos-emp*:
 $0 \leq KL\text{-}div\ \{\}\ a\ e$ ⟨*proof*⟩

lemma *KL-div-pos-gen*:
 fixes $a\ d{::}'b \Rightarrow real$
 assumes *fin*: *finite S*
 assumes *non-null*: $\bigwedge i.\ i{\in}S \Longrightarrow 0 < a\ i\ \bigwedge i.\ i{\in}\ S \Longrightarrow 0 < d\ i$
 assumes *sum-a-one*: $(\sum\ i \in S.\ a\ i) = 1$
 assumes *sum-d-one*: $(\sum\ i \in S.\ d\ i) = 1$
shows $0 \leq KL\text{-}div\ S\ a\ d$
  ⟨*proof*⟩

theorem *KL-div-pos2*:
 fixes $a\ d{::}'b \Rightarrow real$

assumes *fin*: *finite S*
assumes *non-null*: $\bigwedge i.\ i{\in}S \Longrightarrow 0 \leq a\ i$ $\bigwedge i.\ i{\in}\ S \Longrightarrow 0 < d\ i$
assumes *sum-a-one*: $(\sum\ i \in S.\ a\ i) = 1$
assumes *sum-c-one*: $(\sum\ i \in S.\ d\ i) = 1$
shows $0 \leq$ *KL-div S a d*
⟨*proof*⟩

lemma *sum-div-1*:
fixes $f{::}'b \Rightarrow 'c{::}field$
assumes $(\sum i{\in}A.\ f\ i) \neq 0$
shows $(\sum i{\in}A.\ f\ i\ /\ (\sum j{\in}A.\ f\ j)) = 1$
⟨*proof*⟩

theorem *rate-lower-bound*:
shows $\mathcal{H}(X) \leq$ *code-rate enc X*
⟨*proof*⟩

end

end

References

[1] T. M. Cover and J. A. Thomas. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, 2006.