Lovasz Local Lemma

Chelsea Edmonds and Lawrence C. Paulson

April 18, 2024

Abstract

This entry aims to formalise several useful general techniques for using the *probabilistic method* for combinatorial structures (or discrete spaces more generally). In particular, it focuses on bounding tools, such as the union and complete independence bounds, and the first formalisation of the pivotal Lovász local lemma. The formalisation focuses on the general lemma, however also proves several useful variations, including the more well known symmetric version. Both the original formalisation and several of the variations used dependency graphs, which were formalised using Noschinski's general directed graph library [2]. Additionally, the entry provides several useful existence lemmas, required at the end of most probabilistic proofs on combinatorial structures. Finally, the entry includes several significant extensions to the existing probability libraries, particularly for conditional probability (such as Bayes theorem) and independent events. The formalisation is primarily based on Alon and Spencer's textbook [1], as well as Zhao's course notes [3].

Contents

1 Extensional function extras

Counting lemmas (i.e. reasoning on cardinality) of sets on the extensional function relation

theory *PiE-Rel-Extras* **imports** *Card-Partitions.Card-Partitions*
begin

1.1 Relations and Extensional Function sets

A number of lemmas to convert between relations and functions for counting purposes. Note, ultimately not needed in this formalisation, but may be of use in the future

lemma *Range-unfold*: $Range\ r = \{y.\ \exists x.\ (x, y) \in r\}$
⟨*proof*⟩

definition *fun-to-rel*:: $'a\ set \Rightarrow\ 'b\ set\ \Rightarrow ('a \Rightarrow\ 'b) \Rightarrow ('a \times\ 'b)\ set$ **where**
$fun\text{-}to\text{-}rel\ A\ B\ f \equiv \{(a, b) \mid a\ b\ .\ a \in A \wedge b \in B \wedge f\ a = b\}$

definition *rel-to-fun*:: $('a \times\ 'b)\ set \Rightarrow ('a \Rightarrow\ 'b)$ **where**
$rel\text{-}to\text{-}fun\ R \equiv \lambda\ a\ .\ (if\ a \in Domain\ R\ then\ (THE\ b\ .\ (a, b) \in R)\ else\ undefined)$

lemma *fun-to-relI*: $a \in A \Longrightarrow b \in B \Longrightarrow f\ a = b \Longrightarrow (a, b) \in fun\text{-}to\text{-}rel\ A\ B\ f$
⟨*proof*⟩

lemma *fun-to-rel-alt*: $fun\text{-}to\text{-}rel\ A\ B\ f \equiv \{(a, f\ a) \mid a\ b\ .\ a \in A \wedge f\ a \in B\}$
⟨*proof*⟩

lemma *fun-to-relI2*: $a \in A \Longrightarrow f\ a \in B \Longrightarrow (a, f\ a) \in fun\text{-}to\text{-}rel\ A\ B\ f$
⟨*proof*⟩

lemma *rel-to-fun-in*[*simp*]: $a \in Domain\ R \Longrightarrow (rel\text{-}to\text{-}fun\ R)\ a = (THE\ b\ .\ (a,\ b) \in R)$
⟨*proof*⟩

lemma *rel-to-fun-undefined*[*simp*]: $a \notin Domain\ R \Longrightarrow (rel\text{-}to\text{-}fun\ R)\ a = undefined$
⟨*proof*⟩

lemma *single-valued-unique-Dom-iff*: $single\text{-}valued\ R \longleftrightarrow (\forall\ x \in Domain\ R.\ \exists!\ y\ .\ (x,\ y) \in R)$
⟨*proof*⟩

lemma *rel-to-fun-range*:
assumes $single\text{-}valued\ R$
assumes $a \in Domain\ R$
shows $(THE\ b\ .\ (a,\ b) \in R) \in Range\ R$
⟨*proof*⟩

lemma *rel-to-fun-extensional*: $single\text{-}valued\ R \Longrightarrow rel\text{-}to\text{-}fun\ R \in (Domain\ R \rightarrow_E Range\ R)$
⟨*proof*⟩

lemma *single-value-fun-to-rel*: $single\text{-}valued\ (fun\text{-}to\text{-}rel\ A\ B\ f)$
⟨*proof*⟩

lemma *fun-to-rel-domain*:
assumes $f \in A \rightarrow_E B$
shows $Domain\ (fun\text{-}to\text{-}rel\ A\ B\ f) = A$
⟨*proof*⟩

lemma *fun-to-rel-range*:
assumes $f \in A \rightarrow_E B$
shows $Range\ (fun\text{-}to\text{-}rel\ A\ B\ f) \subseteq B$
⟨*proof*⟩

lemma *rel-to-fun-to-rel*:
assumes $f \in A \rightarrow_E B$
shows $rel\text{-}to\text{-}fun\ (fun\text{-}to\text{-}rel\ A\ B\ f) = f$
⟨*proof*⟩

lemma *fun-to-rel-to-fun*:
assumes $single\text{-}valued\ R$
shows $fun\text{-}to\text{-}rel\ (Domain\ R)\ (Range\ R)\ (rel\text{-}to\text{-}fun\ R) = R$
⟨*proof*⟩

lemma *bij-betw-fun-to-rel*:
assumes $f \in A \rightarrow_E B$
shows $bij\text{-}betw\ (\lambda\ a\ .\ (a,\ f\ a))\ A\ (fun\text{-}to\text{-}rel\ A\ B\ f)$
⟨*proof*⟩

lemma *fun-to-rel-indiv-card*:
  assumes $f \in A \rightarrow_E B$
  shows *card* (*fun-to-rel* A B f) = *card* A
  ⟨*proof*⟩

lemma *fun-to-rel-inj*:
  assumes $C \subseteq A \rightarrow_E B$
  shows *inj-on* (*fun-to-rel* A B) C
⟨*proof*⟩

lemma *fun-to-rel-ss*: *fun-to-rel* A B $f \subseteq A \times B$
  ⟨*proof*⟩

lemma *card-fun-to-rel*: $C \subseteq A \rightarrow_E B \Longrightarrow$ *card* C = *card* ((λf . *fun-to-rel* A B f) ‘ C)
  ⟨*proof*⟩

1.2 Cardinality Lemmas

Lemmas to count variations of filtered sets over the extensional function set relation

lemma *card-PiE-filter-range-set*:
  assumes $\bigwedge a.\ a \in A' \Longrightarrow X\ a \in C$
  assumes $A' \subseteq A$
  assumes *finite* A
  shows *card* $\{f \in A \rightarrow_E C\ .\ \forall\ a \in A'\ .\ f\ a = X\ a\}$ = (*card* C)^(*card* A − *card* A')
⟨*proof*⟩

lemma *card-PiE-filter-range-indiv*: $X\ a' \in C \Longrightarrow a' \in A \Longrightarrow$ *finite* $A \Longrightarrow$
    card $\{f \in A \rightarrow_E C\ .\ f\ a' = X\ a'\}$ = (*card* C)^(*card* A − *1*)
  ⟨*proof*⟩

lemma *card-PiE-filter-range-set-const*: $c \in C \Longrightarrow A' \subseteq A \Longrightarrow$ *finite* $A \Longrightarrow$
    card $\{f \in A \rightarrow_E C\ .\ \forall\ a \in A'\ .\ f\ a = c\}$ = (*card* C)^(*card* A − *card* A')
  ⟨*proof*⟩

lemma *card-PiE-filter-range-set-nat*: $c \in \{0..<n\} \Longrightarrow A' \subseteq A \Longrightarrow$ *finite* $A \Longrightarrow$
    card $\{f \in A \rightarrow_E \{0..<n\}\ .\ \forall\ a \in A'\ .\ f\ a = c\}$ = n^(*card* A − *card* A')
  ⟨*proof*⟩

end

2 Digraph extensions

Extensions to the existing library for directed graphs, basically neighborhood

theory *Digraph-Extensions*
  imports

Graph-Theory.Digraph
Graph-Theory.Pair-Digraph
begin

definition (**in** *pre-digraph*) *neighborhood* :: *'a* ⇒ *'a set* **where**
neighborhood u ≡ {*v* ∈ *verts G* . *dominates G u v*}

lemma (**in** *wf-digraph*) *neighborhood-wf*: *neighborhood v* ⊆ *verts G*
⟨*proof*⟩

lemma (**in** *pair-pre-digraph*) *neighborhood-alt*:
neighborhood u = {*v* ∈ *pverts G* . (*u*, *v*) ∈ *parcs G*}
⟨*proof*⟩

lemma (**in** *fin-digraph*) *neighborhood-finite*: *finite* (*neighborhood v*)
⟨*proof*⟩

lemma (**in** *wf-digraph*) *neighborhood-edge-iff*: *y* ∈ *neighborhood x* ⟷ (*x*, *y*) ∈ *arcs-ends G*
⟨*proof*⟩

lemma (**in** *loopfree-digraph*) *neighborhood-self-not*: *v* ∉ (*neighborhood v*)
⟨*proof*⟩

lemma (**in** *nomulti-digraph*) *inj-on-head-out-arcs*: *inj-on* (*head G*) (*out-arcs G u*)
⟨*proof*⟩

lemma (**in** *nomulti-digraph*) *out-degree-neighborhood*: *out-degree G u* = *card* (*neighborhood u*)
⟨*proof*⟩

lemma (**in** *digraph*) *neighborhood-empty-iff*: *out-degree G u* = *0* ⟷ *neighborhood u* = {}
⟨*proof*⟩

end

3 General Event Lemmas

General lemmas for reasoning on events in probability spaces after different operations

theory *Prob-Events-Extras*
imports
HOL−Probability.Probability
PiE-Rel-Extras
begin

context *prob-space*

begin

lemma *prob-sum-Union*:
assumes *measurable*: *finite A A* $\subseteq$ *events disjoint A*
shows *prob* $(\bigcup A\ ) = (\sum e{\in}A.\ prob\ (e))$
$\langle proof \rangle$

lemma *events-inter*:
assumes *finite S*
assumes $S \neq \{\}$
shows $(\bigwedge A.\ A \in S \Longrightarrow A \in events) \Longrightarrow \bigcap S \in events$
$\langle proof \rangle$

lemma *events-union*:
assumes *finite S*
shows $(\bigwedge A.\ A \in S \Longrightarrow A \in events) \Longrightarrow \bigcup S \in events$
$\langle proof \rangle$

lemma *prob-inter-set-lt-elem*: $A \in events \Longrightarrow prob\ (A \cap (\bigcap AS)) \leq prob\ A$
$\langle proof \rangle$

lemma *Inter-event-ss*: *finite* $A \Longrightarrow A \subseteq events \Longrightarrow A \neq \{\} \Longrightarrow \bigcap A \in events$
$\langle proof \rangle$

lemma *prob-inter-ss-lt*:
assumes *finite A*
assumes $A \subseteq events$
assumes $B \neq \{\}$
assumes $B \subseteq A$
shows $prob\ (\bigcap A) \leq prob\ (\bigcap B)$
$\langle proof \rangle$

lemma *prob-inter-ss-lt-index*:
assumes *finite A*
assumes F ‘ $A \subseteq events$
assumes $B \neq \{\}$
assumes $B \subseteq A$
shows $prob\ (\bigcap (F$ ‘ $A)) \leq prob\ (\bigcap (F$ ‘$B))$
$\langle proof \rangle$

lemma *space-compl-double*:
assumes $S \subseteq events$
shows $((-)\ (space\ M))$ ‘ $(((-)\ (space\ M))$ ‘ $S) = S$
$\langle proof \rangle$

lemma *bij-betw-compl-sets*:
assumes $S \subseteq events$
assumes $S' = ((-)\ (space\ M))$ ‘ S
shows *bij-betw* $((-)\ (space\ M))\ S'\ S$

⟨*proof*⟩

lemma *bij-betw-compl-sets-rev*:
  assumes $S \subseteq$ *events*
  assumes $S' = ((-)\ (space\ M))\ `\ S$
  shows *bij-betw* $((-)\ (space\ M))\ S\ S'$
⟨*proof*⟩

lemma *prob0-basic-inter*: $A \in events \Longrightarrow B \in events \Longrightarrow prob\ A = 0 \Longrightarrow prob\ (A \cap B) = 0$
  ⟨*proof*⟩

lemma *prob0-basic-Inter*: $A \in events \Longrightarrow B \subseteq events \Longrightarrow prob\ A = 0 \Longrightarrow prob\ (A \cap (\bigcap B)) = 0$
  ⟨*proof*⟩

lemma *prob1-basic-inter*: $A \in events \Longrightarrow B \in events \Longrightarrow prob\ A = 1 \Longrightarrow prob\ (A \cap B) = prob\ B$
  ⟨*proof*⟩

lemma *prob1-basic-Inter*:
  assumes $A \in events\ B \subseteq events$
  assumes $prob\ A = 1$
  assumes $B \neq \{\}$
  assumes *finite* B
  shows $prob\ (A \cap (\bigcap B)) = prob\ (\bigcap B)$
⟨*proof*⟩

lemma *compl-identity*: $A \in events \Longrightarrow space\ M - (space\ M - A) = A$
  ⟨*proof*⟩

lemma *prob-addition-rule*: $A \in events \Longrightarrow B \in events \Longrightarrow$
    $prob\ (A \cup B) = prob\ A + prob\ B - prob\ (A \cap B)$
  ⟨*proof*⟩

lemma *compl-subset-in-events*: $S \subseteq events \Longrightarrow (-)\ (space\ M)\ `\ S \subseteq events$
  ⟨*proof*⟩

lemma *prob-compl-diff-inter*: $A \in events \Longrightarrow B \in events \Longrightarrow$
    $prob\ (A \cap (space\ M - B)) = prob\ A - prob\ (A \cap B)$
  ⟨*proof*⟩

lemma *bij-betw-prod-prob*: *bij-betw* $f\ A\ B \Longrightarrow (\prod b{\in}B.\ prob\ b) = (\prod a{\in}A.\ prob\ (f\ a))$
  ⟨*proof*⟩

definition *event-compl* :: $'a\ set \Rightarrow\ 'a\ set$ **where**
event-compl $A \equiv space\ M - A$

lemma *compl-Union*: $A \neq \{\} \Longrightarrow$ *space M* $-$ $(\bigcup A) = (\bigcap a \in A$. (*space M* $-$ a))
⟨*proof*⟩

lemma *compl-Union-fn*: $A \neq \{\} \Longrightarrow$ *space M* $-$ $(\bigcup (F$ ‘ $A)) = (\bigcap a \in A$. (*space M* $-$ *F a*))
⟨*proof*⟩

end

Reasoning on the probability of function sets

lemma *card-PiE-val-ss-eq*:
assumes *finite A*
assumes $b \in B$
assumes $d \subseteq A$
assumes $B \neq \{\}$
assumes *finite B*
shows *card* $\{f \in (A \rightarrow_E B)$. $(\forall\ v \in d\ .f\ v = b)\}$/*card* $(A \rightarrow_E B)$ = *1*/((*card B*) *powi* (*card d*))
(**is** *card* $\{f \in ?C$. $(\forall\ v \in d\ .f\ v = b)\}$/*card* $?C$ = *1*/((*card B*) *powi* (*card d*)))
⟨*proof*⟩

lemma *card-PiE-val-indiv-eq*:
assumes *finite A*
assumes $b \in B$
assumes $d \in A$
assumes $B \neq \{\}$
assumes *finite B*
shows *card* $\{f \in (A \rightarrow_E B)$. $f\ d = b\}$/*card* $(A \rightarrow_E B)$ = *1*/(*card B*)
(**is** *card* $\{f \in ?C\ .f\ d = b\}$/*card* $?C$ = *1*/(*card B*))
⟨*proof*⟩

lemma *prob-uniform-ex-fun-space*:
assumes *finite A*
assumes $b \in B$
assumes $d \subseteq A$
assumes $B \neq \{\}$
assumes $A \neq \{\}$
assumes *finite B*
shows *prob-space.prob* (*uniform-count-measure* $(A \rightarrow_E B)$) $\{f \in (A \rightarrow_E B)$. $(\forall\ v \in d\ .f\ v = b)\}$ =
1/((*card B*) *powi* (*card d*))
⟨*proof*⟩

proposition *integrable-uniform-count-measure-finite*:
fixes g :: $'a \Rightarrow\ 'b$::{*banach*, *second-countable-topology*}
shows *finite A* $\Longrightarrow$ *integrable* (*uniform-count-measure A*) *g*
⟨*proof*⟩

end

4 Conditional Probability Library Extensions

theory *Cond-Prob-Extensions*
imports
Prob-Events-Extras
Design-Theory.Multisets-Extras
begin

4.1 Miscellaneous Set and List Lemmas

lemma *nth-image-tl*:
assumes $xs \neq []$
shows $nth\ xs\ `\ \{1..<length\ xs\} = set(tl\ xs)$
⟨*proof*⟩

lemma *exists-list-card*:
assumes *finite S*
obtains *xs* **where** $set\ xs = S$ **and** $length\ xs = card\ S$
⟨*proof*⟩

lemma *bij-betw-inter-empty*:
assumes *bij-betw f A B*
assumes $A' \subseteq A$
assumes $A'' \subseteq A$
assumes $A' \cap A'' = \{\}$
shows $f\ `\ A' \cap f\ `\ A'' = \{\}$
⟨*proof*⟩

lemma *bij-betw-image-comp-eq*:
assumes *bij-betw g T S*
shows $(F \circ g)\ `\ T = F\ `\ S$
⟨*proof*⟩

lemma *prod-card-image-set-eq*:
assumes $bij\text{-}betw\ f\ \{0..<card\ S\}\ S$
assumes *finite S*
shows $(\prod i \in \{n..<(card\ S)\}\ .\ g\ (f\ i)) = (\prod i \in f\ `\ \{n..<card\ S\}\ .\ g\ i)$
⟨*proof*⟩

lemma *set-take-distinct-elem-not*:
assumes *distinct xs*
assumes $i < length\ xs$
shows $xs\ !\ i \notin set\ (take\ i\ xs)$
⟨*proof*⟩

4.2 Conditional Probability Basics

context *prob-space*
begin

Abbreviation to mirror mathematical notations

abbreviation *cond-prob-ev* :: *'a set* ⇒ *'a set* ⇒ *real* ($\mathcal{P}$'(- | -')) **where**
$\mathcal{P}(B \mid A) \equiv \mathcal{P}(x\ in\ M.\ (x \in B) \mid (x \in A))$

lemma *cond-prob-inter*: $\mathcal{P}(B \mid A) = \mathcal{P}(\omega\ in\ M.\ (\omega \in B \cap A))\ /\ \mathcal{P}(\omega\ in\ M.\ (\omega \in A))$
⟨*proof*⟩

lemma *cond-prob-ev-def*:
assumes $A \in$ *events* $B \in$ *events*
shows $\mathcal{P}(B \mid A) =$ *prob* $(A \cap B)$ / *prob* A
⟨*proof*⟩

lemma *measurable-in-ev*:
assumes $A \in$ *events*
shows *Measurable.pred M* $(\lambda\ x\ .\ x \in A)$
⟨*proof*⟩

lemma *measure-uniform-measure-eq-cond-prob-ev*:
assumes $A \in$ *events* $B \in$ *events*
shows $\mathcal{P}(A \mid B) = \mathcal{P}(x\ in$ *uniform-measure* $M\ \{x \in space\ M.\ x \in B\}.\ x \in A)$
⟨*proof*⟩

lemma *measure-uniform-measure-eq-cond-prob-ev2*:
assumes $A \in$ *events* $B \in$ *events*
shows $\mathcal{P}(A \mid B) =$ *measure* (*uniform-measure* $M\ \{x \in space\ M.\ x \in B\}$) A
⟨*proof*⟩

lemma *measure-uniform-measure-eq-cond-prob-ev3*:
assumes $A \in$ *events* $B \in$ *events*
shows $\mathcal{P}(A \mid B) =$ *measure* (*uniform-measure* $M\ B$) A
⟨*proof*⟩

lemma *prob-space-cond-prob-uniform*:
assumes *prob* $(\{x \in space\ M.\ Q\ x\}) > 0$
shows *prob-space* (*uniform-measure* $M\ \{x \in space\ M.\ Q\ x\}$)
⟨*proof*⟩

lemma *prob-space-cond-prob-event*:
assumes *prob* $B > 0$
shows *prob-space* (*uniform-measure* $M\ B$)
⟨*proof*⟩

Note this case shouldn't be used. Conditional probability should have > 0 assumption

lemma *cond-prob-empty*: $\mathcal{P}(B \mid \{\}) = 0$
⟨*proof*⟩

lemma *cond-prob-space*: $\mathcal{P}(A \mid space\ M) = \mathcal{P}(w\ in\ M\ .\ w \in A)$

⟨*proof*⟩

lemma *cond-prob-space-ev*: **assumes** $A \in$ *events* **shows** $\mathcal{P}(A \mid$ *space* $M)$ = *prob* A
⟨*proof*⟩

lemma *cond-prob-UNIV*: $\mathcal{P}(A \mid UNIV) = \mathcal{P}(w$ *in* M . $w \in A)$
⟨*proof*⟩

lemma *cond-prob-UNIV-ev*: $A \in$ *events* $\Longrightarrow \mathcal{P}(A \mid UNIV)$ = *prob* A
⟨*proof*⟩

lemma *cond-prob-neg*:
assumes $A \in$ *events* $B \in$ *events*
assumes *prob* $A > 0$
shows $\mathcal{P}((\mathit{space}\ M - B) \mid A) = 1 - \mathcal{P}(B \mid A)$
⟨*proof*⟩

4.3 Bayes Theorem

lemma *prob-intersect-A*:
assumes $A \in$ *events* $B \in$ *events*
shows *prob* $(A \cap B)$ = *prob* $A * \mathcal{P}(B \mid A)$
⟨*proof*⟩

lemma *prob-intersect-B*:
assumes $A \in$ *events* $B \in$ *events*
shows *prob* $(A \cap B)$ = *prob* $B * \mathcal{P}(A \mid B)$
⟨*proof*⟩

theorem *Bayes-theorem*:
assumes $A \in$ *events* $B \in$ *events*
shows *prob* $B * \mathcal{P}(A \mid B)$ = *prob* $A * \mathcal{P}(B \mid A)$
⟨*proof*⟩

corollary *Bayes-theorem-div*:
assumes $A \in$ *events* $B \in$ *events*
shows $\mathcal{P}(A \mid B) = (\mathit{prob}\ A * \mathcal{P}(B \mid A))/(\mathit{prob}\ B)$
⟨*proof*⟩

lemma *cond-prob-dual-intersect*:
assumes $A \in$ *events* $B \in$ *events* $C \in$ *events*
assumes *prob* $C \neq 0$
shows $\mathcal{P}(A \mid (B \cap C)) = \mathcal{P}(A \cap B \mid C)/ \mathcal{P}(B \mid C)$ (**is** *?LHS* = *?RHS*)
⟨*proof*⟩

lemma *cond-prob-ev-double*:
assumes $A \in$ *events* $B \in$ *events* $C \in$ *events*

assumes *prob C* > *0*
shows $\mathcal{P}$(*x in* (*uniform-measure M C*). (*x* ∈ *A*) | (*x* ∈ *B*)) = $\mathcal{P}$(*A* | (*B* ∩ *C*))
⟨*proof*⟩

lemma *cond-prob-inter-set-lt*:
assumes *A* ∈ *events B* ∈ *events AS* ⊆ *events*
assumes *finite AS*
shows $\mathcal{P}$((*A* ∩ (⋂*AS*)) | *B*) ≤ $\mathcal{P}$(*A*|*B*) (**is** *?LHS* ≤ *?RHS*)
⟨*proof*⟩

4.4 Conditional Probability Multiplication Rule

Many list and indexed variations of this lemma

lemma *prob-cond-Inter-List*:
assumes *xs* ≠ []
assumes ⋀ *A*. *A* ∈ *set xs* ⟹ *A* ∈ *events*
shows *prob* (⋂(*set xs*)) = *prob* (*hd xs*) ∗ (∏ *i* = *1*..<(*length xs*) .
$\mathcal{P}$((*xs* ! *i*) | (⋂(*set* (*take i xs*)))))
⟨*proof*⟩

lemma *prob-cond-Inter-index*:
fixes *n* :: *nat*
assumes *n* > *0*
assumes *F* ' {*0*..<*n*} ⊆ *events*
shows *prob* (⋂ (*F* '{*0*..<*n*})) = *prob* (*F 0*) ∗ (∏ *i* ∈ {*1*..<*n*} .
$\mathcal{P}$(*F i* | (⋂ (*F*'{*0*..<*i*}))))
⟨*proof*⟩

lemma *prob-cond-Inter-index-compl*:
fixes *n* :: *nat*
assumes *n* > *0*
assumes *F* ' {*0*..<*n*} ⊆ *events*
shows *prob* (⋂ *x* ∈ {*0*..<*n*} . *space M* − *F x*) = *prob* (*space M* − *F 0*) ∗ (∏ *i* ∈ {*1*..<*n*} .
$\mathcal{P}$(*space M* − *F i* | (⋂ *j*∈ {*0*..<*i*}. *space M* − *F j*)))
⟨*proof*⟩

lemma *prob-cond-Inter-take-cond*:
assumes *xs* ≠ []
assumes *set xs* ⊆ *events*
assumes *S* ⊆ *events*
assumes *S* ≠ {}
assumes *finite S*
assumes *prob* (⋂*S*) > *0*
shows $\mathcal{P}$((⋂(*set xs*)) | (⋂ *S*)) = (∏ *i* = *0*..<(*length xs*) . $\mathcal{P}$((*xs* ! *i*) | (⋂(*set* (*take i xs*) ∪ *S*))))
⟨*proof*⟩

lemma *prob-cond-Inter-index-cond-set*:
fixes n :: *nat*
assumes $n > 0$
assumes *finite* E
assumes $E \neq \{\}$
assumes $E \subseteq$ *events*
assumes F ‘ $\{0..<n\} \subseteq$ *events*
assumes *prob* $(\bigcap E) > 0$
shows $\mathcal{P}((\bigcap (F\ ‘\ \{0..<n\}))\ |\ (\bigcap E)) = (\prod i \in \{0..<n\}.\ \mathcal{P}(F\ i\ |\ (\bigcap ((F\ ‘\ \{0..<i\}) \cup E))))$
⟨*proof*⟩

lemma *prob-cond-Inter-index-cond-compl-set*:
fixes n :: *nat*
assumes $n > 0$
assumes *finite* E
assumes $E \neq \{\}$
assumes $E \subseteq$ *events*
assumes F ‘ $\{0..<n\} \subseteq$ *events*
assumes *prob* $(\bigcap E) > 0$
shows $\mathcal{P}((\bigcap ((-)\ (space\ M)\ ‘\ F\ ‘\ \{0..<n\}))\ |\ (\bigcap E)) =$
$(\prod i = 0..<n .\ \mathcal{P}((space\ M - F\ i)\ |\ (\bigcap ((-)\ (space\ M)\ ‘\ F\ ‘\ \{0..<i\} \cup E))))$
⟨*proof*⟩

lemma *prob-cond-Inter-index-cond*:
fixes n :: *nat*
assumes $n > 0$
assumes $n < m$
assumes F ‘ $\{0..<m\} \subseteq$ *events*
assumes *prob* $(\bigcap j \in \{n..<m\}.\ F\ j) > 0$
shows $\mathcal{P}((\bigcap (F\ ‘\ \{0..<n\}))\ |\ (\bigcap j \in \{n..<m\}\ .\ F\ j)) = (\prod i \in \{0..<n\}.\ \mathcal{P}(F\ i\ |\ (\bigcap ((F\ ‘\ \{0..<i\}) \cup (F\ ‘\ \{n..<m\})))))$
⟨*proof*⟩

lemma *prob-cond-Inter-index-cond-compl*:
fixes n :: *nat*
assumes $n > 0$
assumes $n < m$
assumes F ‘ $\{0..<m\} \subseteq$ *events*
assumes *prob* $(\bigcap j \in \{n..<m\}.\ F\ j) > 0$
shows $\mathcal{P}((\bigcap ((-)\ (space\ M)\ ‘\ F\ ‘\ \{0..<n\}))\ |\ (\bigcap (\ F\ ‘\ \{n..<m\}))) =$
$(\prod i = 0..<n .\ \mathcal{P}((space\ M - F\ i)\ |\ (\bigcap ((-)\ (space\ M)\ ‘\ F\ ‘\ \{0..<i\} \cup (F\ ‘\ \{n..<m\})))))$
⟨*proof*⟩

lemma *prob-cond-Inter-take-cond-neg*:
assumes $xs \neq []$
assumes *set* $xs \subseteq$ *events*

assumes $S \subseteq$ *events*
assumes $S \neq \{\}$
assumes *finite* S
assumes *prob* $(\bigcap S) > 0$
shows $\mathcal{P}((\bigcap((-)\ (space\ M)\ `\ (set\ xs)))\ |\ (\bigcap\ S)) =$
$(\prod i = 0..<(length\ xs)\ .\ \mathcal{P}((space\ M - xs\ !\ i)\ |\ (\bigcap((-)\ (space\ M)\ `\ (set\ (take\ i\ xs\ ))\ \cup\ S))))$
⟨*proof*⟩

lemma *prob-cond-Inter-List-Index*:
assumes $xs \neq []$
assumes *set* $xs \subseteq$ *events*
shows $prob\ (\bigcap(set\ xs)) = prob\ (hd\ xs) * (\prod i = 1..<(length\ xs)\ .$
$\mathcal{P}((xs\ !\ i)\ |\ (\bigcap\ j \in \{0..<i\}\ .\ xs\ !\ j)))$
⟨*proof*⟩

lemma *obtains-prob-cond-Inter-index*:
assumes $S \neq \{\}$
assumes $S \subseteq$ *events*
assumes *finite* S
obtains *xs* **where** *set* $xs = S$ **and** *length* $xs =$ *card* S **and**
$prob\ (\bigcap S) = prob\ (hd\ xs) * (\prod i = 1..<(length\ xs)\ .\ \mathcal{P}((xs\ !\ i)\ |\ (\bigcap\ j \in \{0..<i\}\ .\ xs\ !\ j)))$
⟨*proof*⟩

lemma *obtain-list-index*:
assumes *bij-betw* $g\ \{0..<card\ S\}\ S$
assumes *finite* S
obtains *xs* **where** *set* $xs = S$ **and** $\bigwedge i\ .\ i \in \{0..<card\ S\} \Longrightarrow g\ i = xs\ !\ i$ **and** *distinct xs*
⟨*proof*⟩

lemma *prob-cond-inter-fn*:
assumes *bij-betw* $g\ \{0..<card\ S\}\ S$
assumes *finite* S
assumes $S \neq \{\}$
assumes $S \subseteq$ *events*
shows $prob\ (\bigcap S) = prob\ (g\ 0) * (\prod i \in \{1..<(card\ S)\}\ .\ \mathcal{P}(g\ i\ |\ (\bigcap(g\ `\ \{0..<i\}))))$
⟨*proof*⟩

lemma *prob-cond-inter-obtain-fn*:
assumes $S \neq \{\}$
assumes $S \subseteq$ *events*
assumes *finite* S
obtains f **where** *bij-betw* $f\ \{0..<card\ S\}\ S$ **and**
$prob\ (\bigcap S) = prob\ (f\ 0) * (\prod i \in \{1..<(card\ S)\}\ .\ \mathcal{P}(f\ i\ |\ (\bigcap(f\ `\ \{0..<i\}))))$
⟨*proof*⟩

lemma *prob-cond-inter-obtain-fn-compl*:
assumes $S \neq \{\}$
assumes $S \subseteq events$
assumes *finite* S
obtains f **where** *bij-betw* f $\{0..<card\ S\}$ S **and** *prob* $(\bigcap((-)\ (space\ M)\ `\ S))$ $=$
prob $(space\ M - f\ 0) * (\prod i \in \{1..<(card\ S)\}\ .\ \mathcal{P}(space\ M - f\ i \mid (\bigcap((-)\ (space\ M)\ `\ f\ `\ \{0..<i\}))))$
⟨*proof*⟩

lemma *prob-cond-Inter-index-cond-fn*:
assumes $I \neq \{\}$
assumes *finite* I
assumes *finite* E
assumes $E \neq \{\}$
assumes $E \subseteq events$
assumes $F\ `\ I \subseteq events$
assumes *prob* $(\bigcap E) > 0$
assumes *bb*: *bij-betw* g $\{0..<card\ I\}$ I
shows $\mathcal{P}((\bigcap(F\ `\ g\ `\ \{0..<card\ I\})) \mid (\bigcap E)) =$
$(\prod i \in \{0..<card\ I\}.\ \ \mathcal{P}(F\ (g\ i) \mid (\bigcap((F\ `\ g`\ \{0..<i\}) \cup E))))$
⟨*proof*⟩

lemma *prob-cond-Inter-index-cond-obtains*:
assumes $I \neq \{\}$
assumes *finite* I
assumes *finite* E
assumes $E \neq \{\}$
assumes $E \subseteq events$
assumes $F\ `\ I \subseteq events$
assumes *prob* $(\bigcap E) > 0$
obtains g **where** *bij-betw* g $\{0..<card\ I\}$ I **and** $\mathcal{P}((\bigcap(F\ `\ g\ `\ \{0..<card\ I\})) \mid (\bigcap E)) =$
$(\prod i \in \{0..<card\ I\}.\ \ \mathcal{P}(F\ (g\ i) \mid (\bigcap((F\ `\ g`\ \{0..<i\}) \cup E))))$
⟨*proof*⟩

lemma *prob-cond-Inter-index-cond-compl-fn*:
assumes $I \neq \{\}$
assumes *finite* I
assumes *finite* E
assumes $E \neq \{\}$
assumes $E \subseteq events$
assumes $F\ `\ I \subseteq events$
assumes *prob* $(\bigcap E) > 0$
assumes *bb*: *bij-betw* g $\{0..<card\ I\}$ I
shows $\mathcal{P}((\bigcap Aj \in I\ .\ space\ M - F\ Aj) \mid (\bigcap E)) =$
$(\prod i \in \{0..<card\ I\}.\ \ \mathcal{P}(space\ M - F\ (g\ i) \mid (\bigcap(((\lambda Aj.\ space\ M - F\ Aj)\ `\ g\ `\ \{0..<i\}) \cup E))))$
⟨*proof*⟩

lemma *prob-cond-Inter-index-cond-compl-obtains*:
assumes $I \neq \{\}$
assumes *finite I*
assumes *finite E*
assumes $E \neq \{\}$
assumes $E \subseteq events$
assumes F ‘ $I \subseteq events$
assumes $prob\ (\bigcap E) > 0$
obtains g **where** *bij-betw g* $\{0..<card\ I\}$ I **and** $\mathcal{P}((\bigcap Aj \in I\ .\ space\ M - F\ Aj) \mid (\bigcap E)) =$
$(\prod i \in \{0..<card\ I\}.\ \mathcal{P}(space\ M - F\ (g\ i) \mid (\bigcap(((\lambda Aj.\ space\ M - F\ Aj)\ \text{‘}\ g\ \text{‘}\ \{0..<i\}) \cup E))))$
⟨*proof*⟩

lemma *prob-cond-inter-index-fn2*:
assumes F ‘ $S \subseteq events$
assumes *finite S*
assumes $card\ S > 0$
assumes *bij-betw g* $\{0..<card\ S\}$ S
shows $prob\ (\bigcap(F\ \text{‘}S)) = prob\ (F\ (g\ 0)) * (\prod i \in \{1..<(card\ S)\}\ .\ \mathcal{P}(F\ (g\ i) \mid (\bigcap(F\ \text{‘}\ g\ \text{‘}\ \{0..<i\}))))$
⟨*proof*⟩

lemma *prob-cond-inter-index-fn*:
assumes F ‘ $S \subseteq events$
assumes *finite S*
assumes $S \neq \{\}$
assumes *bij-betw g* $\{0..<card\ S\}$ S
shows $prob\ (\bigcap(F\ \text{‘}S)) = prob\ (F\ (g\ 0)) * (\prod i \in \{1..<(card\ S)\}\ .\ \mathcal{P}(F\ (g\ i) \mid (\bigcap(F\ \text{‘}\ g\ \text{‘}\ \{0..<i\}))))$
⟨*proof*⟩

lemma *prob-cond-inter-index-obtain-fn*:
assumes F ‘ $S \subseteq events$
assumes *finite S*
assumes $S \neq \{\}$
obtains g **where** *bij-betw g* $\{0..<card\ S\}$ S **and**
$prob\ (\bigcap(F\ \text{‘}S)) = prob\ (F\ (g\ 0)) * (\prod i \in \{1..<(card\ S)\}\ .\ \mathcal{P}(F\ (g\ i) \mid (\bigcap(F\ \text{‘}\ g\ \text{‘}\ \{0..<i\}))))$
⟨*proof*⟩

lemma *prob-cond-inter-index-fn-compl*:
assumes $S \neq \{\}$
assumes F ‘ $S \subseteq events$
assumes *finite S*
assumes *bij-betw f* $\{0..<card\ S\}$ S
shows $prob\ (\bigcap((-)\ (space\ M)\ \text{‘}\ F\ \text{‘}\ S)) = prob\ (space\ M - F\ (f\ 0)) *$
$(\prod i \in \{1..<(card\ S)\}\ .\ \mathcal{P}(space\ M - F\ (f\ i) \mid (\bigcap((-)\ (space\ M)\ \text{‘}\ F\ \text{‘}\ f\ \text{‘}$

$\{0..<i\})))$
$\langle proof \rangle$

lemma *prob-cond-inter-index-obtain-fn-compl*:
assumes $S \neq \{\}$
assumes F ‘ $S \subseteq events$
assumes *finite S*
obtains *f* **where** $bij\text{-}betw\ f\ \{0..<card\ S\}\ S$ **and**
$prob\ (\bigcap((-)\ (space\ M)\ ‘\ F\ ‘\ S)) = prob\ (space\ M - F\ (f\ 0)) *$
$(\prod i \in \{1..<(card\ S)\}\ .\ \mathcal{P}(space\ M - F\ (f\ i) \mid (\bigcap((-)\ (space\ M)\ ‘\ F\ ‘\ f\ ‘$
$\{0..<i\}))))$
$\langle proof \rangle$

lemma *prob-cond-Inter-take*:
assumes $S \neq \{\}$
assumes $S \subseteq events$
assumes *finite S*
obtains *xs* **where** $set\ xs = S$ **and** $length\ xs = card\ S$ **and**
$prob\ (\bigcap S) = prob\ (hd\ xs) * (\prod i = 1..<(length\ xs)\ .\ \mathcal{P}((xs\ !\ i) \mid (\bigcap(set\ (take\ i$
$xs\ )))))$
$\langle proof \rangle$

lemma *prob-cond-Inter-set-bound*:
assumes $A \neq \{\}$
assumes $A \subseteq events$
assumes *finite A*
assumes $\bigwedge\ Ai\ .\ f\ Ai \geq 0 \wedge f\ Ai \leq 1$
assumes $\bigwedge Ai\ S.\ Ai \in A \Longrightarrow S \subseteq A - \{Ai\} \Longrightarrow S \neq \{\} \Longrightarrow \mathcal{P}(Ai \mid (\bigcap S)) \geq f$
Ai
assumes $\bigwedge\ Ai.\ Ai \in A \Longrightarrow prob\ Ai \geq f\ Ai$
shows $prob\ (\bigcap A) \geq (\prod a' \in A\ .\ f\ a')$
$\langle proof \rangle$
end

end

5 Independent Events

theory *Indep-Events* **imports** *Cond-Prob-Extensions*
begin

5.1 More bijection helpers

lemma *bij-betw-obtain-subsetr*:
assumes *bij-betw f A B*
assumes $A' \subseteq A$
obtains B' **where** $B' \subseteq B$ **and** $B' = f$ ‘ A'

⟨*proof*⟩

lemma *bij-betw-obtain-subsetl*:
assumes *bij-betw f A B*
assumes $B' \subseteq B$
obtains A' **where** $A' \subseteq A$ **and** $B' = f$ ‘ A'
⟨*proof*⟩

lemma *bij-betw-remove*: *bij-betw f A B* $\Longrightarrow a \in A \Longrightarrow$ *bij-betw f* $(A - \{a\})$ $(B - \{f\ a\})$
⟨*proof*⟩

5.2 Independent Event Extensions

Extensions on both the indep_event definition and the indep_events definition

context *prob-space*
begin

lemma *indep-eventsD*: *indep-events A I* $\Longrightarrow$ $(A\text{‘}I \subseteq events) \Longrightarrow J \subseteq I \Longrightarrow J \neq \{\} \Longrightarrow$ *finite* $J \Longrightarrow$
prob $(\bigcap j\in J.\ A\ j) = (\prod j\in J.\ prob\ (A\ j))$
⟨*proof*⟩

lemma
assumes *indep*: *indep-event A B*
shows *indep-eventD-ev1*: $A \in events$
and *indep-eventD-ev2*: $B \in events$
⟨*proof*⟩

lemma *indep-eventD*:
assumes *ie*: *indep-event A B*
shows *prob* $(A \cap B) = prob\ (A) * prob\ (B)$
⟨*proof*⟩

lemma *indep-eventI*[*intro*]:
assumes *ev*: $A \in events$ $B \in events$
and *indep*: *prob* $(A \cap B) = prob\ A * prob\ B$
shows *indep-event A B*
⟨*proof*⟩

Alternate set definition - when no possibility of duplicate objects

definition *indep-events-set* :: *'a set set* ⇒ *bool* **where**
indep-events-set $E \equiv (E \subseteq events \wedge (\forall J.\ J \subseteq E \longrightarrow finite\ J \longrightarrow J \neq \{\} \longrightarrow prob\ (\bigcap J) = (\prod i\in J.\ prob\ i)))$

lemma *indep-events-setI*[*intro*]: $E \subseteq events \Longrightarrow (\bigwedge J.\ J \subseteq E \Longrightarrow finite\ J \Longrightarrow J \neq \{\} \Longrightarrow$
prob $(\bigcap J) = (\prod i\in J.\ prob\ i)) \Longrightarrow$ *indep-events-set E*

⟨*proof*⟩

lemma *indep-events-subset*:
indep-events-set $E \longleftrightarrow (\forall J \subseteq E.\ \textit{indep-events-set}\ J)$
⟨*proof*⟩

lemma *indep-events-subset2*:
indep-events-set $E \Longrightarrow J \subseteq E \Longrightarrow \textit{indep-events-set}\ J$
⟨*proof*⟩

lemma *indep-events-set-events*: *indep-events-set* $E \Longrightarrow (\bigwedge e.\ e \in E \Longrightarrow e \in \textit{events})$

⟨*proof*⟩

lemma *indep-events-set-events-ss*: *indep-events-set* $E \Longrightarrow E \subseteq \textit{events}$
⟨*proof*⟩

lemma *indep-events-set-probs*: *indep-events-set* $E \Longrightarrow J \subseteq E \Longrightarrow \textit{finite}\ J \Longrightarrow J \neq \{\} \Longrightarrow$
$\textit{prob}\ (\bigcap J) = (\prod i \in J.\ \textit{prob}\ i)$
⟨*proof*⟩

lemma *indep-events-set-prod-all*: *indep-events-set* $E \Longrightarrow \textit{finite}\ E \Longrightarrow E \neq \{\} \Longrightarrow$
$\textit{prob}\ (\bigcap E) = \textit{prod}\ \textit{prob}\ E$
⟨*proof*⟩

lemma *indep-events-not-contain-compl*:
assumes *indep-events-set* E
assumes $A \in E$
assumes $\textit{prob}\ A > 0$ $\textit{prob}\ A < 1$
shows $(\textit{space}\ M - A) \notin E$ (**is** $?A' \notin E$)
⟨*proof*⟩

lemma *indep-events-contain-compl-prob01*:
assumes *indep-events-set* E
assumes $A \in E$
assumes $\textit{space}\ M - A \in E$
shows $\textit{prob}\ A = 0 \lor \textit{prob}\ A = 1$
⟨*proof*⟩

lemma *indep-events-set-singleton*:
assumes $A \in \textit{events}$
shows *indep-events-set* $\{A\}$
⟨*proof*⟩

lemma *indep-events-pairs*:
assumes *indep-events-set* S
assumes $A \in S$ $B \in S$ $A \neq B$

shows *indep-event A B*
⟨*proof*⟩

lemma *indep-events-inter-pairs*:
assumes *indep-events-set S*
assumes *finite A finite B*
assumes $A \neq \{\}$ $B \neq \{\}$
assumes $A \subseteq S$ $B \subseteq S$ $A \cap B = \{\}$
shows *indep-event* $(\bigcap A)$ $(\bigcap B)$
⟨*proof*⟩

lemma *indep-events-inter-single*:
assumes *indep-events-set S*
assumes *finite B*
assumes $B \neq \{\}$
assumes $A \in S$ $B \subseteq S$ $A \notin B$
shows *indep-event A* $(\bigcap B)$
⟨*proof*⟩

lemma *indep-events-set-prob1*:
assumes $A \in$ *events*
assumes *prob* $A = 1$
assumes $A \notin S$
assumes *indep-events-set S*
shows *indep-events-set* $(S \cup \{A\})$
⟨*proof*⟩

lemma *indep-events-set-prob0*:
assumes $A \in$ *events*
assumes *prob* $A = 0$
assumes $A \notin S$
assumes *indep-events-set S*
shows *indep-events-set* $(S \cup \{A\})$
⟨*proof*⟩

lemma *indep-event-commute*:
assumes *indep-event A B*
shows *indep-event B A*
⟨*proof*⟩

Showing complement operation maintains independence

lemma *indep-event-one-compl*:
assumes *indep-event A B*
shows *indep-event A* (*space M* $-$ *B*)
⟨*proof*⟩

lemma *indep-event-one-compl-rev*:
assumes $B \in$ *events*

assumes *indep-event A* (*space M* − *B*)
shows *indep-event A B*
⟨*proof*⟩

lemma *indep-event-double-compl*: *indep-event A B* ⟹ *indep-event* (*space M* − *A*) (*space M* − *B*)
⟨*proof*⟩

lemma *indep-event-double-compl-rev*: *A* ∈ *events* ⟹ *B* ∈ *events* ⟹ *indep-event* (*space M* − *A*) (*space M* − *B*) ⟹ *indep-event A B*
⟨*proof*⟩

lemma *indep-events-set-one-compl*:
assumes *indep-events-set S*
assumes *A* ∈ *S*
shows *indep-events-set* ({*space M* − *A*} ∪ (*S* − {*A*}))
⟨*proof*⟩

lemma *indep-events-set-update-compl*:
assumes *indep-events-set E*
assumes *E* = *A* ∪ *B*
assumes *A* ∩ *B* = {}
assumes *finite E*
shows *indep-events-set* (((−) (*space M*) ‘ *A*) ∪ *B*)
⟨*proof*⟩

lemma *indep-events-set-compl*:
assumes *indep-events-set E*
assumes *finite E*
shows *indep-events-set* ((λ *e*. *space M* − *e*) ‘ *E*)
⟨*proof*⟩

lemma *indep-event-empty*:
assumes *A* ∈ *events*
shows *indep-event A* {}
⟨*proof*⟩

lemma *indep-event-compl-inter*:
assumes *indep-event A C*
assumes *B* ∈ *events*
assumes *indep-event A* (*B* ∩ *C*)
shows *indep-event A* ((*space M* − *B*) ∩ *C*)
⟨*proof*⟩

lemma *indep-events-index-subset*:
indep-events F E ⟷ (∀ *J*⊆*E*. *indep-events F J*)

⟨*proof*⟩

lemma *indep-events-index-subset2*:
indep-events $F\ E \Longrightarrow J \subseteq E \Longrightarrow$ *indep-events* $F\ J$
⟨*proof*⟩

lemma *indep-events-events-ss*: *indep-events* $F\ E \Longrightarrow\ F\ `\ E \subseteq$ *events*
⟨*proof*⟩

lemma *indep-events-events*: *indep-events* $F\ E \Longrightarrow (\bigwedge e.\ e \in E \Longrightarrow F\ e \in$ *events*$)$
⟨*proof*⟩

lemma *indep-events-probs*: *indep-events* $F\ E \Longrightarrow J \subseteq E \Longrightarrow$ *finite* $J \Longrightarrow J \neq \{\}$ $\Longrightarrow$ *prob* $(\bigcap(F\ `\ J)) = (\prod i{\in}J.$ *prob* $(F\ i))$
⟨*proof*⟩

lemma *indep-events-prod-all*: *indep-events* $F\ E \Longrightarrow$ *finite* $E \Longrightarrow E \neq \{\} \Longrightarrow$ *prob* $(\bigcap(F\ `\ E)) = (\prod i{\in}E.$ *prob* $(F\ i))$
⟨*proof*⟩

lemma *indep-events-ev-not-contain-compl*:
assumes *indep-events* $F\ E$
assumes $A \in E$
assumes *prob* $(F\ A) > 0$ *prob* $(F\ A) < 1$
shows (*space* $M - F\ A) \notin F\ `\ E$ (**is** $?A' \notin F\ `\ E$)
⟨*proof*⟩

lemma *indep-events-singleton*:
assumes $F\ A \in$ *events*
shows *indep-events* $F\ \{A\}$
⟨*proof*⟩

lemma *indep-events-ev-pairs*:
assumes *indep-events* $F\ S$
assumes $A \in S\ B \in S\ A \neq B$
shows *indep-event* $(F\ A)\ (F\ B)$
⟨*proof*⟩

lemma *indep-events-ev-inter-pairs*:
assumes *indep-events* $F\ S$
assumes *finite* A *finite* B
assumes $A \neq \{\}\ B \neq \{\}$
assumes $A \subseteq S\ B \subseteq S\ A \cap B = \{\}$
shows *indep-event* $(\bigcap(F\ `\ A))\ (\bigcap(F\ `\ B))$
⟨*proof*⟩

lemma *indep-events-ev-inter-single*:

assumes *indep-events F S*
assumes *finite B*
assumes $B \neq \{\}$
assumes $A \in S$ $B \subseteq S$ $A \notin B$
shows *indep-event* $(F\ A)$ $(\bigcap (F\ `\ B))$
⟨*proof*⟩

lemma *indep-events-fn-eq*:
assumes $\bigwedge$ *Ai*. $Ai \in E \Longrightarrow F\ Ai = G\ Ai$
assumes *indep-events F E*
shows *indep-events G E*
⟨*proof*⟩

lemma *indep-events-fn-eq-iff*:
assumes $\bigwedge$ *Ai*. $Ai \in E \Longrightarrow F\ Ai = G\ Ai$
shows *indep-events F E* $\longleftrightarrow$ *indep-events G E*
⟨*proof*⟩

lemma *indep-events-one-compl*:
assumes *indep-events F S*
assumes $A \in S$
shows *indep-events* (λ *i*. *if* $(i = A)$ *then* (*space M* − *F i*) *else F i*) *S* (**is** *indep-events ?G S*)
⟨*proof*⟩

lemma *indep-events-update-compl*:
assumes *indep-events F E*
assumes $E = A \cup B$
assumes $A \cap B = \{\}$
assumes *finite E*
shows *indep-events* (λ *Ai*. *if* $(Ai \in A)$ *then* (*space M* − (*F Ai*)) *else* (*F Ai*)) *E*
⟨*proof*⟩

lemma *indep-events-compl*:
assumes *indep-events F E*
assumes *finite E*
shows *indep-events* (λ *Ai*. *space M* − *F Ai*) *E*
⟨*proof*⟩

lemma *indep-events-impl-inj-on*:
assumes *finite A*
assumes *indep-events F A*
assumes $\bigwedge$ A' . $A' \in A \Longrightarrow$ *prob* $(F\ A') > 0 \land$ *prob* $(F\ A') < 1$
shows *inj-on F A*
⟨*proof*⟩

lemma *indep-events-imp-set*:
assumes *finite A*
assumes *indep-events F A*

assumes $\bigwedge A' . A' \in A \Longrightarrow prob\ (F\ A') > 0 \wedge prob\ (F\ A') < 1$
shows *indep-events-set* $(F\ `\ A)$
⟨*proof*⟩

lemma *indep-event-set-equiv-bij*:
assumes *bij-betw* $F\ A\ E$
assumes *finite* E
shows *indep-events-set* $E \longleftrightarrow$ *indep-events* $F\ A$
⟨*proof*⟩

5.3 Mutual Independent Events

Note, set based version only if no duplicates in usage case. The mutual_indep_events definition is more general and recommended

definition *mutual-indep-set*:: $'a\ set \Rightarrow 'a\ set\ set \Rightarrow bool$
where *mutual-indep-set* $A\ S \longleftrightarrow A \in events \wedge S \subseteq events \wedge (\forall\ T \subseteq S\ .\ T \neq \{\} \longrightarrow prob\ (A \cap (\bigcap T)) = prob\ A * prob\ (\bigcap T))$

lemma *mutual-indep-setI*[*intro*]: $A \in events \Longrightarrow S \subseteq events \Longrightarrow (\bigwedge T.\ T \subseteq S \Longrightarrow T \neq \{\} \Longrightarrow$
$prob\ (A \cap (\bigcap T)) = prob\ A * prob\ (\bigcap T)) \Longrightarrow$ *mutual-indep-set* $A\ S$
⟨*proof*⟩

lemma *mutual-indep-setD*[*dest*]: *mutual-indep-set* $A\ S \Longrightarrow T \subseteq S \Longrightarrow T \neq \{\} \Longrightarrow prob\ (A \cap (\bigcap T)) = prob\ A * prob\ (\bigcap T)$
⟨*proof*⟩

lemma *mutual-indep-setD2*[*dest*]: *mutual-indep-set* $A\ S \Longrightarrow A \in events$
⟨*proof*⟩

lemma *mutual-indep-setD3*[*dest*]: *mutual-indep-set* $A\ S \Longrightarrow S \subseteq events$
⟨*proof*⟩

lemma *mutual-indep-subset*: *mutual-indep-set* $A\ S \Longrightarrow T \subseteq S \Longrightarrow$ *mutual-indep-set* $A\ T$
⟨*proof*⟩

lemma *mutual-indep-event-set-defD*:
assumes *mutual-indep-set* $A\ S$
assumes *finite* T
assumes $T \subseteq S$
assumes $T \neq \{\}$
shows *indep-event* $A\ (\bigcap T)$
⟨*proof*⟩

lemma *mutual-indep-event-defI*: $A \in events \Longrightarrow S \subseteq events \Longrightarrow (\bigwedge T.\ T \subseteq S \Longrightarrow T \neq \{\} \Longrightarrow$

indep-event A $(\bigcap T)) \Longrightarrow$ *mutual-indep-set A S*
⟨*proof*⟩

lemma *mutual-indep-singleton-event*: *mutual-indep-set A S* $\Longrightarrow B \in S \Longrightarrow$ *indep-event A B*
⟨*proof*⟩

lemma *mutual-indep-cond*:
assumes $A \in$ *events* **and** $T \subseteq$ *events* **and** *finite T*
and *mutual-indep-set A S* **and** $T \subseteq S$ **and** $T \neq \{\}$ **and** *prob* $(\bigcap T) \neq 0$
shows $\mathcal{P}(A \mid (\bigcap T)) =$ *prob A*
⟨*proof*⟩

lemma *mutual-indep-cond-full*:
assumes $A \in$ *events* **and** $S \subseteq$ *events* **and** *finite S*
and *mutual-indep-set A S* **and** $S \neq \{\}$ **and** *prob* $(\bigcap S) \neq 0$
shows $\mathcal{P}(A \mid (\bigcap S)) =$ *prob A*
⟨*proof*⟩

lemma *mutual-indep-cond-single*:
assumes $A \in$ *events* **and** $B \in$ *events*
and *mutual-indep-set A S* **and** $B \in S$ **and** *prob* $B \neq 0$
shows $\mathcal{P}(A \mid B) =$ *prob A*
⟨*proof*⟩

lemma *mutual-indep-set-empty*: $A \in$ *events* $\Longrightarrow$ *mutual-indep-set A* $\{\}$
⟨*proof*⟩

lemma *not-mutual-indep-set-itself*:
assumes *prob* $A > 0$ **and** *prob* $A < 1$
shows $\neg$ *mutual-indep-set A* $\{A\}$
⟨*proof*⟩

lemma *is-mutual-indep-set-itself*:
assumes $A \in$ *events*
assumes *prob* $A = 0 \vee$ *prob* $A = 1$
shows *mutual-indep-set A* $\{A\}$
⟨*proof*⟩

lemma *mutual-indep-set-singleton*:
assumes *indep-event A B*
shows *mutual-indep-set A* $\{B\}$
⟨*proof*⟩

lemma *mutual-indep-set-one-compl*:
assumes *mutual-indep-set A S*
assumes *finite S*
assumes $B \in S$
shows *mutual-indep-set A* $(\{$*space M* $- B\} \cup S)$

⟨*proof*⟩

lemma *mutual-indep-events-set-update-compl*:
assumes *mutual-indep-set X E*
assumes $E = A \cup B$
assumes $A \cap B = \{\}$
assumes *finite E*
shows *mutual-indep-set X* $(((-)\ (\mathit{space}\ M)\ `\ A) \cup B)$
⟨*proof*⟩

lemma *mutual-indep-events-compl*:
assumes *finite S*
assumes *mutual-indep-set A S*
shows *mutual-indep-set A* $((\lambda\ s\ .\ \mathit{space}\ M - s)\ `\ S)$
⟨*proof*⟩

lemma *mutual-indep-set-all*:
assumes $A \subseteq \mathit{events}$
assumes $\bigwedge Ai.\ Ai \in A \Longrightarrow (\mathit{mutual\text{-}indep\text{-}set}\ Ai\ (A - \{Ai\}))$
shows *indep-events-set A*
⟨*proof*⟩

Prefered version using indexed notation

definition *mutual-indep-events*:: $'a\ \mathit{set} \Rightarrow (\mathit{nat} \Rightarrow 'a\ \mathit{set}) \Rightarrow \mathit{nat}\ \mathit{set} \Rightarrow \mathit{bool}$
where *mutual-indep-events A F I* $\longleftrightarrow A \in \mathit{events} \wedge (F\ `\ I \subseteq \mathit{events}) \wedge (\forall\ J \subseteq I\ .\ J \neq \{\} \longrightarrow \mathit{prob}\ (A \cap (\bigcap j \in J\ .\ F\ j)) = \mathit{prob}\ A * \mathit{prob}\ (\bigcap j \in J\ .\ F\ j))$

lemma *mutual-indep-eventsI*[*intro*]: $A \in \mathit{events} \Longrightarrow (F\ `\ I \subseteq \mathit{events}) \Longrightarrow (\bigwedge J.\ J \subseteq I \Longrightarrow J \neq \{\} \Longrightarrow$
$\mathit{prob}\ (A \cap (\bigcap j \in J\ .\ F\ j)) = \mathit{prob}\ A * \mathit{prob}\ (\bigcap j \in J\ .\ F\ j)) \Longrightarrow$ *mutual-indep-events A F I*
⟨*proof*⟩

lemma *mutual-indep-eventsD*[*dest*]: *mutual-indep-events A F I* $\Longrightarrow J \subseteq I \Longrightarrow J \neq \{\} \Longrightarrow \mathit{prob}\ (A \cap (\bigcap j \in J\ .\ F\ j)) = \mathit{prob}\ A * \mathit{prob}\ (\bigcap j \in J\ .\ F\ j)$
⟨*proof*⟩

lemma *mutual-indep-eventsD2*[*dest*]: *mutual-indep-events A F I* $\Longrightarrow A \in \mathit{events}$
⟨*proof*⟩

lemma *mutual-indep-eventsD3*[*dest*]: *mutual-indep-events A F I* $\Longrightarrow F\ `\ I \subseteq \mathit{events}$
⟨*proof*⟩

lemma *mutual-indep-ev-subset*: *mutual-indep-events A F I* $\Longrightarrow J \subseteq I \Longrightarrow$ *mutual-indep-events A F J*
⟨*proof*⟩

lemma *mutual-indep-event-defD*:
assumes *mutual-indep-events A F I*
assumes *finite J*
assumes $J \subseteq I$
assumes $J \neq \{\}$
shows *indep-event A* $(\bigcap j \in J\ .\ F\ j)$
⟨*proof*⟩

lemma *mutual-ev-indep-event-defI*: $A \in events \Longrightarrow F\ `\ I \subseteq events \Longrightarrow (\bigwedge J.\ J \subseteq I \Longrightarrow J \neq \{\} \Longrightarrow$
indep-event A $(\bigcap (F`\ J))) \Longrightarrow$ *mutual-indep-events A F I*
⟨*proof*⟩

lemma *mutual-indep-ev-singleton-event*:
assumes *mutual-indep-events A F I*
assumes $B \in F\ `\ I$
shows*indep-event A B*
⟨*proof*⟩

lemma *mutual-indep-ev-singleton-event2*:
assumes *mutual-indep-events A F I*
assumes $i \in I$
shows*indep-event A* $(F\ i)$
⟨*proof*⟩

lemma *mutual-indep-iff*:
shows *mutual-indep-events A F I* $\longleftrightarrow$ *mutual-indep-set A* $(F\ `\ I)$
⟨*proof*⟩

lemma *mutual-indep-ev-cond*:
assumes $A \in events$ **and** $F\ `\ J \subseteq events$ **and** *finite J*
and *mutual-indep-events A F I* **and** $J \subseteq I$ **and** $J \neq \{\}$ **and** *prob* $(\bigcap (F\ `J)) \neq 0$
shows $\mathcal{P}(A\ |(\bigcap (F\ `\ J))) = prob\ A$
⟨*proof*⟩

lemma *mutual-indep-ev-cond-full*:
assumes $A \in events$ **and** $F\ `\ I \subseteq events$ **and** *finite I*
and *mutual-indep-events A F I* **and** $I \neq \{\}$ **and** *prob* $(\bigcap (F\ `\ I)) \neq 0$
shows $\mathcal{P}(A\ |(\bigcap (F\ `I))) = prob\ A$
⟨*proof*⟩

lemma *mutual-indep-ev-cond-single*:
assumes $A \in events$ **and** $B \in events$
and *mutual-indep-events A F I* **and** $B \in F\ `\ I$ **and** *prob* $B \neq 0$
shows $\mathcal{P}(A\ |B) = prob\ A$
⟨*proof*⟩

lemma *mutual-indep-ev-empty*: $A \in events \Longrightarrow$ *mutual-indep-events A F* $\{\}$
⟨*proof*⟩

lemma *not-mutual-indep-ev-itself*:
assumes *prob* $A > 0$ **and** *prob* $A < 1$ **and** $A = F\ i$
shows $\neg$ *mutual-indep-events* $A\ F\ \{i\}$
⟨*proof*⟩

lemma *is-mutual-indep-ev-itself*:
assumes $A \in$ *events* **and** $A = F\ i$
assumes *prob* $A = 0 \vee$ *prob* $A = 1$
shows *mutual-indep-events* $A\ F\ \{i\}$
⟨*proof*⟩

lemma *mutual-indep-ev-singleton*:
assumes *indep-event* $A\ (F\ i)$
shows *mutual-indep-events* $A\ F\ \{i\}$
⟨*proof*⟩

lemma *mutual-indep-ev-one-compl*:
assumes *mutual-indep-events* $A\ F\ I$
assumes *finite* I
assumes $i \in I$
assumes *space* $M - F\ i = F\ j$
shows *mutual-indep-events* $A\ F\ (\{j\} \cup\ I)$
⟨*proof*⟩

lemma *mutual-indep-events-update-compl*:
assumes *mutual-indep-events* $X\ F\ S$
assumes $S = A \cup B$
assumes $A \cap B = \{\}$
assumes *finite* S
assumes *bij-betw* $G\ A\ A'$
assumes $\bigwedge i.\ i \in A \Longrightarrow F\ (G\ i) =$ *space* $M - F\ i$
shows *mutual-indep-events* $X\ F\ (A' \cup B)$
⟨*proof*⟩

lemma *mutual-indep-ev-events-compl*:
assumes *finite* S
assumes *mutual-indep-events* $A\ F\ S$
assumes *bij-betw* $G\ S\ S'$
assumes $\bigwedge i.\ i \in S \Longrightarrow F\ (G\ i) =$ *space* $M - F\ i$
shows *mutual-indep-events* $A\ F\ S'$
⟨*proof*⟩

Important lemma on relation between independence and mutual independence of a set

lemma *mutual-indep-ev-set-all*:
assumes F ‘ $I \subseteq$ *events*
assumes $\bigwedge i.\ i \in I \Longrightarrow$ (*mutual-indep-events* $(F\ i)\ F\ (I - \{i\})$)
shows *indep-events* $F\ I$

⟨*proof*⟩

end
end

6 The Basic Probabilistic Method Framework

This theory includes all aspects of step (3) and (4) of the basic method framework, which are purely probabilistic

theory *Basic-Method* **imports** *Indep-Events*
begin

6.1 More Set and Multiset lemmas

lemma *card-size-set-mset*: $card\ (set\text{-}mset\ A) \leq size\ A$
⟨*proof*⟩

lemma *Union-exists*: $\{a \in A\ .\ \exists\ b \in B\ .\ P\ a\ b\} = (\bigcup b \in B\ .\ \{a \in A\ .\ P\ a\ b\})$
⟨*proof*⟩

lemma *Inter-forall*: $B \neq \{\} \Longrightarrow \{a \in A\ .\ \forall\ b \in B\ .\ P\ a\ b\} = (\bigcap b \in B\ .\ \{a \in A\ .\ P\ a\ b\})$
⟨*proof*⟩

lemma *function-map-multi-filter-size*:
assumes $image\text{-}mset\ F\ (mset\text{-}set\ A) = B$ **and** $finite\ A$
shows $card\ \{a \in A\ .\ P\ (F\ a)\} = size\ \{\#\ b \in\#\ B\ .\ P\ b\ \#\}$
⟨*proof*⟩

lemma *bij-mset-obtain-set-elem*:
assumes $image\text{-}mset\ F\ (mset\text{-}set\ A) = B$
assumes $b \in\#\ B$
obtains a **where** $a \in A$ **and** $F\ a = b$
⟨*proof*⟩

lemma *bij-mset-obtain-mset-elem*:
assumes $finite\ A$
assumes $image\text{-}mset\ F\ (mset\text{-}set\ A) = B$
assumes $a \in A$
obtains b **where** $b \in\#\ B$ **and** $F\ a = b$
⟨*proof*⟩

lemma *prod-fn-le1*:
fixes $f :: 'c \Rightarrow ('d :: \{comm\text{-}monoid\text{-}mult,\ linordered\text{-}semidom\})$
assumes $finite\ A$
assumes $A \neq \{\}$
assumes $\bigwedge y.\ y \in A \Longrightarrow f\ y \geq 0 \land f\ y < 1$
shows $(\prod x \in A.\ f\ x) < 1$

⟨*proof*⟩

context *prob-space*
begin

6.2 Existence Lemmas

lemma *prob-lt-one-obtain*:
assumes $\{e \in \mathit{space}\ M\ .\ Q\ e\} \in \mathit{events}$
assumes $\mathit{prob}\ \{e \in \mathit{space}\ M\ .\ Q\ e\} < 1$
obtains e **where** $e \in \mathit{space}\ M$ **and** $\neg\ Q\ e$
⟨*proof*⟩

lemma *prob-gt-zero-obtain*:
assumes $\{e \in \mathit{space}\ M\ .\ Q\ e\} \in \mathit{events}$
assumes $\mathit{prob}\ \{e \in \mathit{space}\ M\ .\ Q\ e\} > 0$
obtains e **where** $e \in \mathit{space}\ M$ **and** $Q\ e$
⟨*proof*⟩

lemma *inter-gt0-event*:
assumes $F\ `\ I \subseteq \mathit{events}$
assumes $\mathit{prob}\ (\bigcap\ i \in I\ .\ (\mathit{space}\ M - (F\ i))) > 0$
shows $(\bigcap\ i \in I\ .\ (\mathit{space}\ M - (F\ i))) \in \mathit{events}$ **and** $(\bigcap\ i \in I\ .\ (\mathit{space}\ M - (F\ i))) \neq \{\}$
⟨*proof*⟩

lemma *obtain-intersection*:
assumes $F\ `\ I \subseteq \mathit{events}$
assumes $\mathit{prob}\ (\bigcap\ i \in I\ .\ (\mathit{space}\ M - (F\ i)))> 0$
obtains e **where** $e \in \mathit{space}\ M$ **and** $\bigwedge\ i.\ i \in I \Longrightarrow e \notin F\ i$
⟨*proof*⟩

lemma *obtain-intersection-prop*:
assumes $F\ `\ I \subseteq \mathit{events}$
assumes $\bigwedge\ i.\ i \in I \Longrightarrow F\ i = \{e \in \mathit{space}\ M\ .\ P\ e\ i\}$
assumes $\mathit{prob}\ (\bigcap\ i \in I\ .\ (\mathit{space}\ M - (F\ i)))> 0$
obtains e **where** $e \in \mathit{space}\ M$ **and** $\bigwedge\ i.\ i \in I \Longrightarrow \neg\ P\ e\ i$
⟨*proof*⟩

lemma *not-in-big-union*:
assumes $\bigwedge\ i\ .\ i \in A \Longrightarrow e \notin i$
shows $e \notin (\bigcup A)$
⟨*proof*⟩

lemma *not-in-big-union-fn*:
assumes $\bigwedge\ i\ .\ i \in A \Longrightarrow e \notin F\ i$
shows $e \notin (\bigcup i \in A\ .\ F\ i)$
⟨*proof*⟩

lemma *obtain-intersection-union*:
  assumes F ' $I \subseteq$ *events*
  assumes *prob* ($\bigcap$ $i \in I$. (*space* M − (F i)))> *0*
  obtains e **where** $e \in$ *space* M **and** $e \notin$ ($\bigcup i \in I$. F i)
⟨*proof*⟩

6.3 Basic Bounds

Lemmas on the Complete Independence and Union bound

lemma *complete-indep-bound1*:
  assumes *finite* A
  assumes $A \neq \{\}$
  assumes $A \subseteq$ *events*
  assumes *indep-events-set* A
  assumes $\bigwedge$ a . $a \in A \Longrightarrow$ *prob* a < *1*
  shows *prob* (*space* M − ($\bigcap A$)) > *0*
⟨*proof*⟩

lemma *complete-indep-bound1-index*:
  assumes *finite* A
  assumes $A \neq \{\}$
  assumes F ' $A \subseteq$ *events*
  assumes *indep-events* F A
  assumes $\bigwedge$ a . $a \in A \Longrightarrow$ *prob* (F a) < *1*
  shows *prob* (*space* M − ($\bigcap$(F' A))) > *0*
⟨*proof*⟩

lemma *complete-indep-bound2*:
  assumes *finite* A
  assumes $A \subseteq$ *events*
  assumes *indep-events-set* A
  assumes $\bigwedge$ a . $a \in A \Longrightarrow$ *prob* a < *1*
  shows *prob* (*space* M − ($\bigcup A$)) > *0*
⟨*proof*⟩

lemma *complete-indep-bound2-index*:
  assumes *finite* A
  assumes F ' $A \subseteq$ *events*
  assumes *indep-events* F A
  assumes $\bigwedge$ a . $a \in A \Longrightarrow$ *prob* (F a) < *1*
  shows *prob* (*space* M − ($\bigcup$(F ' A))) > *0*
⟨*proof*⟩

lemma *complete-indep-bound3*:
  assumes *finite* A
  assumes $A \neq \{\}$
  assumes F ' $A \subseteq$ *events*
  assumes *indep-events* F A
  assumes $\bigwedge$ a . $a \in A \Longrightarrow$ *prob* (F a) < *1*

shows *prob* ($\bigcap a \in A.$ *space M* $-$ *F a*) > 0
⟨*proof*⟩

Combining complete independence with existence step

lemma *complete-indep-bound-obtain*:
assumes *finite A*
assumes $A \subseteq$ *events*
assumes *indep-events-set A*
assumes $\bigwedge a . a \in A \Longrightarrow$ *prob a* < 1
obtains *e* **where** $e \in$ *space M* **and** $e \notin \bigcup A$
⟨*proof*⟩

lemma *Union-bound-events*:
assumes *finite A*
assumes $A \subseteq$ *events*
shows *prob* ($\bigcup A$) $\leq$ ($\sum a \in A.$ *prob a*)
⟨*proof*⟩

lemma *Union-bound-events-fun*:
assumes *finite A*
assumes *f ‘ A* $\subseteq$ *events*
shows *prob* ($\bigcup$(*f ‘ A*)) $\leq$ ($\sum a \in A.$ *prob* (*f a*))
⟨*proof*⟩

lemma *Union-bound-avoid*:
assumes *finite A*
assumes ($\sum a \in A.$ *prob a*) < 1
assumes $A \subseteq$ *events*
shows *prob* (*space M* $- \bigcup A$) > 0
⟨*proof*⟩

lemma *Union-bound-avoid-fun*:
assumes *finite A*
assumes ($\sum a \in A.$ *prob* (*f a*)) < 1
assumes *f‘A* $\subseteq$ *events*
shows *prob* (*space M* $- \bigcup$(*f ‘ A*)) > 0
⟨*proof*⟩

Combining union bound with existance step

lemma *Union-bound-obtain*:
assumes *finite A*
assumes ($\sum a \in A.$ *prob a*) < 1
assumes $A \subseteq$ *events*
obtains *e* **where** $e \in$ *space M* **and** $e \notin \bigcup A$
⟨*proof*⟩

lemma *Union-bound-obtain-fun*:
assumes *finite A*

assumes $(\sum a \in A.\ prob\ (f\ a)) < 1$
assumes $f\ `\ A \subseteq events$
obtains e **where** $e \in space\ M$ **and** $e \notin \bigcup(\ f`\ A)$
⟨*proof*⟩

lemma *Union-bound-obtain-compl*:
assumes *finite A*
assumes $(\sum a \in A.\ prob\ a) < 1$
assumes $A \subseteq events$
obtains e **where** $e \in (space\ M - \bigcup A)$
⟨*proof*⟩

lemma *Union-bound-obtain-compl-fun*:
assumes *finite A*
assumes $(\sum a \in A.\ prob\ (f\ a)) < 1$
assumes $f\ `\ A \subseteq events$
obtains e **where** $e \in (space\ M - \bigcup(\ f`\ A))$
⟨*proof*⟩

end

end

7 Lovasz Local Lemma

theory *Lovasz-Local-Lemma*
imports
Basic-Method
HOL−Real-Asymp.Real-Asymp
Indep-Events
Digraph-Extensions
begin

7.1 Random Lemmas on Product Operator

lemma *prod-constant-ge*:
fixes $y :: 'b :: \{comm\text{-}monoid\text{-}mult,\ linordered\text{-}semidom\}$
assumes $card\ A \le k$
assumes $y \ge 0$ **and** $y < 1$
shows $(\prod x{\in}A.\ y) \ge y\ \hat{}\ k$
⟨*proof*⟩

lemma (**in** *linordered-idom*) *prod-mono3*:
assumes $finite\ J\ I \subseteq J\ \bigwedge i.\ i \in J \Longrightarrow 0 \le f\ i\ \ (\bigwedge i.\ i \in J \Longrightarrow f\ i \le 1)$
shows $prod\ f\ J \le prod\ f\ I$
⟨*proof*⟩

lemma *bij-on-ss-image*:
assumes $A \subseteq B$

assumes *bij-betw g B B'*
shows *g ' A ⊆ B'*
⟨*proof*⟩

lemma *bij-on-ss-proper-image*:
assumes *A ⊂ B*
assumes *bij-betw g B B'*
shows *g ' A ⊂ B'*
⟨*proof*⟩

7.2 Dependency Graph Concept

Uses directed graphs. The pair_digraph locale was sufficient as multi-edges are irrelevant

locale *dependency-digraph* = *pair-digraph G* :: *nat pair-pre-digraph* + *prob-space M* :: *'a measure*
for *G M* + **fixes** *F* :: *nat ⇒ 'a set*
assumes *vss*: *F ' (pverts G) ⊆ events*
assumes *mis*: ⋀ *i*. *i ∈ (pverts G) ⟹ mutual-indep-events (F i) F ((pverts G) − ({i} ∪ neighborhood i))*
begin

lemma *dep-graph-indiv-nh-indep*:
assumes *A ∈ pverts G B ∈ pverts G*
assumes *B ∉ neighborhood A*
assumes *A ≠ B*
assumes *prob (F B) ≠ 0*
shows *𝒫((F A) | (F B)) = prob (F A)*
⟨*proof*⟩

lemma *mis-subset*:
assumes *i ∈ pverts G*
assumes *A ⊆ pverts G*
shows *mutual-indep-events (F i) F (A − ({i} ∪ neighborhood i))*
⟨*proof*⟩

lemma *dep-graph-indep-events*:
assumes *A ⊆ pverts G*
assumes ⋀ *Ai*. *Ai ∈ A ⟹ out-degree G Ai = 0*
shows *indep-events F A*
⟨*proof*⟩

end

7.3 Lovasz Local General Lemma

context *prob-space*
begin

lemma *compl-sets-index*:
  assumes F ‘ $A \subseteq$ *events*
  shows $(\lambda\ i.\ space\ M - F\ i)$ ‘ $A \subseteq$ *events*
⟨*proof*⟩

lemma *lovasz-inductive-base*:
  assumes *dependency-digraph G M F*
  assumes $\bigwedge Ai\ .\ Ai \in A \Longrightarrow g\ Ai \geq 0 \wedge g\ Ai < 1$
  assumes $\bigwedge Ai.\ Ai \in A \Longrightarrow (prob\ (F\ Ai) \leq (g\ Ai) * (\prod\ Aj \in pre\text{-}digraph.neighborhood\ G\ Ai.\ (1 - (g\ Aj))))$
  assumes $Ai \in A$
  assumes $pverts\ G = A$
  shows $prob\ (F\ Ai) \leq g\ Ai$
⟨*proof*⟩

lemma *lovasz-inductive-base-set*:
  assumes $N \subseteq A$
  assumes $\bigwedge Ai\ .\ Ai \in A \Longrightarrow g\ Ai \geq 0 \wedge g\ Ai < 1$
  assumes $\bigwedge Ai.\ Ai \in A \Longrightarrow (prob\ (F\ Ai) \leq (g\ Ai) * (\prod\ Aj \in N.\ (1 - (g\ Aj))))$
  assumes $Ai \in A$
  shows $prob\ (F\ Ai) \leq g\ Ai$
⟨*proof*⟩

lemma *split-prob-lt-helper*:
  assumes *dep-graph*: *dependency-digraph G M F*
  assumes *dep-graph-verts*: $pverts\ G = A$
  assumes *fbounds*: $\bigwedge i\ .\ i \in A \Longrightarrow f\ i \geq 0 \wedge f\ i < 1$
  assumes *prob-Ai*: $\bigwedge Ai.\ Ai \in A \Longrightarrow prob\ (F\ Ai) \leq$
    $(f\ Ai) * (\prod\ Aj \in pre\text{-}digraph.neighborhood\ G\ Ai\ .\ (1 - (f\ Aj)))$
  assumes *aiin*: $Ai \in A$
  assumes $N \subseteq pre\text{-}digraph.neighborhood\ G\ Ai$
  assumes $\exists\ P1\ P2.\ \mathcal{P}(F\ Ai \mid \bigcap Aj{\in}S.\ space\ M - F\ Aj) = P1/P2 \wedge$
    $P1 \leq prob\ (F\ Ai) \wedge P2 \geq (\prod\ Aj \in N\ .\ (1 - (f\ Aj)))$
  shows $\mathcal{P}(F\ Ai \mid \bigcap Aj{\in}S.\ space\ M - F\ Aj) \leq f\ Ai$
⟨*proof*⟩

lemma *lovasz-inequality*:
  assumes *finS*: *finite S*
  assumes *sevents*: F ‘ $S \subseteq$ *events*
  assumes *S-subset*: $S \subseteq A - \{Ai\}$
  assumes *prob2*: $prob\ (\bigcap\ Aj \in S\ .\ (space\ M - (F\ Aj))) > 0$
  assumes *irange*: $i \in \{0..<card\ S1\}$
  assumes *bb*: $bij\text{-}betw\ g\ \{0..<card\ S1\}\ S1$
  assumes *s1-def*: $S1 = (S \cap N)$
  assumes *s2-def*: $S2 = S - S1$
  assumes *ne-cond*: $i > 0 \vee S2 \neq \{\}$
  assumes *hyps*: $\bigwedge B.\ B \subset S \Longrightarrow g\ i \in A \Longrightarrow B \subseteq A - \{g\ i\} \Longrightarrow B \neq \{\} \Longrightarrow$
    $0 < prob\ (\bigcap Aj{\in}B.\ space\ M - F\ Aj) \Longrightarrow \mathcal{P}(F\ (g\ i) \mid \bigcap Aj{\in}B.\ space\ M - F\ Aj) \leq f\ (g\ i)$

shows $\mathcal{P}((space\ M - F\ (g\ i)) \mid (\bigcap\ ((\lambda\ i.\ space\ M - F\ i)\ `\ g\ `\ \{0..<i\} \cup ((\lambda\ i.\ space\ M - F\ i)\ `\ S2))))$
$\geq (1 - f\ (g\ i))$
⟨*proof*⟩

The main helper lemma

lemma *lovasz-inductive*:
assumes *finA*: $finite\ A$
assumes *Aevents*: $F\ `\ A \subseteq events$
assumes *fbounds*: $\bigwedge\ i\ .\ i \in A \Longrightarrow f\ i \geq 0 \wedge f\ i < 1$
assumes *dep-graph*: *dependency-digraph G M F*
assumes *dep-graph-verts*: $pverts\ G = A$
assumes *prob-Ai*: $\bigwedge\ Ai.\ Ai \in A \Longrightarrow prob\ (F\ Ai) \leq$
$(f\ Ai) * (\prod\ Aj \in pre\text{-}digraph.neighborhood\ G\ Ai\ .\ (1 - (f\ Aj)))$
assumes *Ai-in*: $Ai \in A$
assumes *S-subset*: $S \subseteq A - \{Ai\}$
assumes *S-nempty*: $S \neq \{\}$
assumes *prob2*: $prob\ (\bigcap\ Aj \in S\ .\ (space\ M - (F\ Aj))) > 0$
shows $\mathcal{P}((F\ Ai) \mid (\bigcap\ Aj \in S\ .\ (space\ M - (F\ Aj)))) \leq f\ Ai$
⟨*proof*⟩

The main lemma

theorem *lovasz-local-general*:
assumes $A \neq \{\}$
assumes $F\ `\ A \subseteq events$
assumes $finite\ A$
assumes $\bigwedge\ Ai\ .\ Ai \in A \Longrightarrow\ f\ Ai \geq 0 \wedge f\ Ai < 1$
assumes *dependency-digraph G M F*
assumes $\bigwedge Ai.\ Ai \in A \Longrightarrow (prob\ (F\ Ai) \leq (f\ Ai) * (\prod\ Aj \in pre\text{-}digraph.neighborhood\ G\ Ai.\ (1 - (f\ Aj))))$
assumes $pverts\ G = A$
shows $prob\ (\bigcap\ Ai \in A\ .\ (space\ M - (F\ Ai))) \geq (\prod\ Ai \in A\ .\ (1 - f\ Ai))\ (\prod\ Ai \in A\ .\ (1 - f\ Ai)) > 0$
⟨*proof*⟩

7.4 Lovasz Corollaries and Variations

corollary *lovasz-local-general-positive*:
assumes $A \neq \{\}$
assumes $F\ `\ A \subseteq events$
assumes $finite\ A$
assumes $\bigwedge\ Ai\ .\ Ai \in A \Longrightarrow\ f\ Ai \geq 0 \wedge f\ Ai < 1$
assumes *dependency-digraph G M F*
assumes $\bigwedge\ Ai.\ Ai \in A \Longrightarrow (prob\ (F\ Ai) \leq$
$(f\ Ai) * (\prod\ Aj \in pre\text{-}digraph.neighborhood\ G\ Ai.\ (1 - (f\ Aj))))$
assumes $pverts\ G = A$
shows $prob\ (\bigcap\ Ai \in A\ .\ (space\ M - (F\ Ai))) > 0$
⟨*proof*⟩

theorem *lovasz-local-symmetric-dep-graph*:
  fixes e :: *real*
  fixes d :: *nat*
  assumes $A \neq \{\}$
  assumes F ‘ $A \subseteq$ *events*
  assumes *finite* A
  assumes *dependency-digraph* G M F
  assumes $\bigwedge Ai.\ \ Ai \in A \Longrightarrow$ *out-degree* $G\ Ai \leq d$
  assumes $\bigwedge Ai.\ Ai \in A \Longrightarrow$ *prob* $(F\ Ai) \leq p$
  assumes $exp(1) *\ p * (d + 1) \leq 1$
  assumes *pverts* $G = A$
  shows *prob* $(\bigcap\ Ai \in A\ .\ (space\ M\ -\ (F\ Ai))) > 0$
⟨*proof*⟩

corollary *lovasz-local-symmetric4gt*:
  fixes e :: *real*
  fixes d :: *nat*
  assumes $A \neq \{\}$
  assumes F ‘ $A \subseteq$ *events*
  assumes *finite* A
  assumes *dependency-digraph* G M F
  assumes $\bigwedge Ai.\ \ Ai \in A \Longrightarrow$ *out-degree* $G\ Ai \leq d$
  assumes $\bigwedge Ai.\ Ai \in A \Longrightarrow$ *prob* $(F\ Ai) \leq p$
  assumes $4 * p * d \leq 1$
  assumes $d \geq 3$
  assumes *pverts* $G = A$
  shows *prob* $(\bigcap\ Ai \in A\ .\ (space\ M\ -\ F\ Ai)) > 0$
⟨*proof*⟩

lemma *lovasz-local-symmetric4*:
  fixes e :: *real*
  fixes d :: *nat*
  assumes $A \neq \{\}$
  assumes F ‘ $A \subseteq$ *events*
  assumes *finite* A
  assumes *dependency-digraph* G M F
  assumes $\bigwedge Ai.\ \ Ai \in A \Longrightarrow$ *out-degree* $G\ Ai \leq d$
  assumes $\bigwedge Ai.\ Ai \in A \Longrightarrow$ *prob* $(F\ Ai) \leq p$
  assumes $4 * p * d \leq 1$
  assumes $d \geq 1$
  assumes *pverts* $G = A$
  shows *prob* $(\bigcap\ Ai \in A\ .\ (space\ M\ -\ F\ Ai)) > 0$
⟨*proof*⟩

Converting between dependency graph and indexed set representation of mutual independence

lemma (**in** *pair-digraph*) *g-Ai-simplification*:
  assumes $Ai \in A$

assumes $g\ Ai \subseteq A - \{Ai\}$
assumes *pverts* $G = A$
assumes *parcs* $G = \{e \in A \times A\ .\ snd\ e \in (A - (\{fst\ e\} \cup (g\ (fst\ e))))\}$
shows $g\ Ai = A - (\{Ai\} \cup neighborhood\ Ai)$
⟨*proof*⟩

lemma *define-dep-graph-set*:
assumes $A \neq \{\}$
assumes $F\ `\ A \subseteq events$
assumes *finite* A
assumes $\bigwedge Ai.\ Ai \in A \Longrightarrow g\ Ai \subseteq A - \{Ai\} \wedge mutual\text{-}indep\text{-}events\ (F\ Ai)\ F\ (g\ Ai)$
shows *dependency-digraph* $(\!|\ pverts = A,\ parcs = \{e \in A \times A\ .\ snd\ e \in (A - (\{fst\ e\} \cup (g\ (fst\ e))))\}\ |\!)\ M\ F$
(**is** *dependency-digraph ?G M F*)
⟨*proof*⟩

lemma *define-dep-graph-deg-bound*:
assumes $A \neq \{\}$
assumes $F\ `\ A \subseteq events$
assumes *finite* A
assumes $\bigwedge Ai.\ Ai \in A \Longrightarrow g\ Ai \subseteq A - \{Ai\} \wedge card\ (g\ Ai) \geq card\ A - d - 1 \wedge$
$mutual\text{-}indep\text{-}events\ (F\ Ai)\ F\ (g\ Ai)$
shows $\bigwedge Ai.\ Ai \in A \Longrightarrow$
$out\text{-}degree\ (\!|\ pverts = A,\ parcs = \{e \in A \times A\ .\ snd\ e \in (A - (\{fst\ e\} \cup (g\ (fst\ e))))\}\ |\!)\ Ai \leq d$
(**is** $\bigwedge Ai.\ Ai \in A \Longrightarrow out\text{-}degree\ (with\text{-}proj\ ?G)\ Ai \leq d$)
⟨*proof*⟩

lemma *obtain-dependency-graph*:
assumes $A \neq \{\}$
assumes $F\ `\ A \subseteq events$
assumes *finite* A
assumes $\bigwedge Ai.\ Ai \in A \Longrightarrow$
$(\exists\ S\ .\ S \subseteq A - \{Ai\} \wedge card\ S \geq card\ A - d - 1 \wedge mutual\text{-}indep\text{-}events\ (F\ Ai)\ F\ S)$
obtains G **where** *dependency-digraph G M F* $pverts\ G = A$ $\bigwedge Ai.\ Ai \in A \Longrightarrow out\text{-}degree\ G\ Ai \leq d$
⟨*proof*⟩

This is the variation of the symmetric version most commonly in use

theorem *lovasz-local-symmetric*:
fixes $d :: nat$
assumes $A \neq \{\}$
assumes $F\ `\ A \subseteq events$
assumes *finite* A
assumes $\bigwedge Ai.\ Ai \in A \Longrightarrow (\exists\ S\ .\ S \subseteq A - \{Ai\} \wedge card\ S \geq card\ A - d - 1 \wedge mutual\text{-}indep\text{-}events\ (F\ Ai)\ F\ S)$

assumes $\bigwedge$ *Ai*. *Ai* $\in$ *A* $\Longrightarrow$ *prob* (*F Ai*) $\leq$ *p*
assumes *exp*(*1*)$*$ *p* $*$ (*d* + *1*) $\leq$ *1*
shows *prob* ($\bigcap$ *Ai* $\in$ *A* . (*space M* $-$ (*F Ai*))) > *0*
⟨*proof*⟩

lemma *lovasz-local-symmetric4-set*:
fixes *d* :: *nat*
assumes *A* $\neq$ {}
assumes *F* ' *A* $\subseteq$ *events*
assumes *finite A*
assumes $\bigwedge$ *Ai*. *Ai* $\in$ *A* $\Longrightarrow$ ($\exists$ *S* . *S* $\subseteq$ *A* $-$ {*Ai*} $\wedge$ *card S* $\geq$ *card A* $-$ *d* $-$ *1*
$\wedge$ *mutual-indep-events* (*F Ai*) *F S*)
assumes $\bigwedge$ *Ai*. *Ai* $\in$ *A* $\Longrightarrow$ *prob* (*F Ai*) $\leq$ *p*
assumes *4* $*$ *p* $*$ *d* $\leq$ *1*
assumes *d* $\geq$ *1*
shows *prob* ($\bigcap$ *Ai* $\in$ *A* . (*space M* $-$ *F Ai*)) > *0*
⟨*proof*⟩
end

end
theory *Lovasz-Local-Root*
imports
PiE-Rel-Extras
Digraph-Extensions

Prob-Events-Extras
Cond-Prob-Extensions
Indep-Events

Basic-Method
Lovasz-Local-Lemma
begin
end

References

[1] N. Alon and J. H. Spencer. *The Probabilistic Method*. Wiley-Interscience Series in Discrete Mathematics and Optimization. Wiley, Hoboken, N.J, 4th edition, 2016.

[2] L. Noschinski. A Graph Library for Isabelle. *Mathematics in Computer Science*, 9(1):23–39, Mar. 2015.

[3] Y. Zhao. Probabilistic methods in combinatorics, 2020. Lecture notes MIT 18.226, Fall 2020, https://ocw.mit.edu/courses/18-226-probabilistic-method-in-combinatorics-fall-2020/resources/mit18_226f20_full_notes/.