# Spivey's Generalized Recurrence for Bell Numbers 

Lukas Bulwahn

September 13, 2023


#### Abstract

This entry defines the Bell numbers [1] as the cardinality of set partitions for a carrier set of given size, and derives Spivey's generalized recurrence relation for Bell numbers [2] following his elegant and intuitive combinatorial proof.

As the set construction for the combinatorial proof requires construction of three intermediate structures, the main difficulty of the formalization is handling the overall combinatorial argument in a structured way. The introduced proof structure allows us to compose the combinatorial argument from its subparts, and supports to keep track how the detailed proof steps are related to the overall argument. To obtain this structure, this entry uses set monad notation for the set construction's definition, introduces suitable predicates and rules, and follows a repeating structure in its Isar proof.


## Contents

1 Bell Numbers and Spivey's Generalized Recurrence ..... 1
1.1 Preliminaries ..... 2
1.1.1 Additions to FuncSet ..... 2
1.1.2 Additions for Injectivity Proofs ..... 2
1.2 Definition of Bell Numbers ..... 3
1.3 Construction of the Partitions ..... 4
1.4 Injectivity of the Set Construction ..... 8
1.5 The Generalized Bell Recurrence Relation ..... 9
1.6 Corollaries of the Generalized Bell Recurrence ..... 12
1.7 Code equations for the computation of Bell numbers ..... 13
1 Bell Numbers and Spivey's Generalized Recur- rence

theory Bell-Numbers imports HOL-Library.FuncSet

HOL-Library.Monad-Syntax
HOL-Library.Code-Target-Nat
HOL-Combinatorics.Stirling
Card-Partitions.Injectivity-Solver
Card-Partitions.Card-Partitions
begin

### 1.1 Preliminaries

### 1.1.1 Additions to FuncSet

lemma extensional-funcset-ext:
assumes $f \in A \rightarrow_{E} B g \in A \rightarrow_{E} B$
assumes $\bigwedge x . x \in A \Longrightarrow f x=g x$
shows $f=g$
using assms by (metis PiE-iff extensionalityI)

### 1.1.2 Additions for Injectivity Proofs

lemma inj-on-impl-inj-on-image: assumes inj-on $f A$
assumes $\bigwedge x . x \in X \Longrightarrow x \subseteq A$ shows inj-on ((`)f) $X$
using assms by (meson inj-onI inj-on-image-eq-iff)
lemma injectivity-union:
assumes $A \cup B=C \cup D$
assumes $P A P C$
assumes $Q B Q D$
$\wedge S T . P S \Longrightarrow Q T \Longrightarrow S \cap T=\{ \}$
shows $A=C \wedge B=D$
using assms Int-Un-distrib Int-commute inf-sup-absorb by blast+
lemma injectivity-image:
assumes $f$ ' $A=g^{\prime} A$
assumes $\forall x \in A$. invert $(f x)=x \wedge \operatorname{invert}(g x)=x$
shows $\forall x \in A . f x=g x$
using assms by (metis (no-types, lifting) image-iff)
lemma injectivity-image-union:
assumes $(\lambda X . X \cup F X)^{\prime} P=(\lambda X . X \cup G X)^{\prime} P^{\prime}$
assumes $\forall X \in P . X \subseteq A \forall X \in P^{\prime} . X \subseteq A$
assumes $\forall X \in P . \forall y \in F X . y \notin A \forall X \in P^{\prime} . \forall y \in G X . y \notin A$
shows $P=P^{\prime}$
proof
show $P \subseteq P^{\prime}$
proof
fix $X$
assume $X \in P$
from assms(1) this obtain $X^{\prime}$ where $X^{\prime} \in P^{\prime}$ and $X \cup F X=X^{\prime} \cup G X^{\prime}$

```
        by (metis imageE image-eqI)
    moreover from assms(2,4) <X\inP> have X:(X\cupFX)\capA=X by auto
    moreover from assms(3,5)<\mp@subsup{X}{}{\prime}\in\mp@subsup{P}{}{\prime}>\mathrm{ have }\mp@subsup{X}{}{\prime}:(\mp@subsup{X}{}{\prime}\cupG\mp@subsup{X}{}{\prime})\capA=\mp@subsup{X}{}{\prime}\mathrm{ by}
auto
    ultimately have }X=\mp@subsup{X}{}{\prime}\mathrm{ by simp
    from this }\langle\mp@subsup{X}{}{\prime}\in\mp@subsup{P}{}{\prime}\rangle\mathrm{ show }X\in\mp@subsup{P}{}{\prime}\mathrm{ by auto
    qed
next
    show }\mp@subsup{P}{}{\prime}\subseteq
    proof
        fix }\mp@subsup{X}{}{\prime
        assume }\mp@subsup{X}{}{\prime}\in\mp@subsup{P}{}{\prime
        from assms(1) this obtain X where X }\inP\mathrm{ and X }\cupFX=\mp@subsup{X}{}{\prime}\cupG\mp@subsup{X}{}{\prime
            by (metis imageE image-eqI)
    moreover from assms(2,4) <X \inP> have X:(X\cupF X) \capA=X by auto
    moreover from assms(3,5)\langle\mp@subsup{X}{}{\prime}\in\mp@subsup{P}{}{\prime}>\mathrm{ have }\mp@subsup{X}{}{\prime}:(\mp@subsup{X}{}{\prime}\cupG\mp@subsup{X}{}{\prime})\capA=\mp@subsup{X}{}{\prime}\mathrm{ by}
auto
    ultimately have X= X' by simp
    from this }\langleX\inP\rangle\mathrm{ show }\mp@subsup{X}{}{\prime}\inP\mathrm{ by auto
    qed
qed
```


### 1.2 Definition of Bell Numbers

definition Bell :: nat $\Rightarrow$ nat
where
Bell $n=$ card $\{P$. partition-on $\{0 . .<n\} P\}$
lemma Bell-altdef:
assumes finite $A$
shows Bell $(\operatorname{card} A)=\operatorname{card}\{P$. partition-on $A P\}$
proof -
from 〈finite $A\rangle$ obtain $f$ where bij: bij-betw $f\{0 . .<\operatorname{card} A\} A$
using ex-bij-betw-nat-finite by blast
from this have inj: inj-on $f\{0 . .<\operatorname{card} A\}$ using bij-betw-imp-inj-on by blast
from bij have image-f-eq: $A=f$ ' $\{0 . .<\operatorname{card} A\}$ using bij-betw-imp-surj-on by blast
have $\forall x \in\{P$. partition-on $\{0 . .<\operatorname{card} A\} P\} . x \subseteq \operatorname{Pow}\{0 . .<\operatorname{card} A\}$
by (auto elim: partition-onE)
from this inj have inj-on $\left(\left({ }^{\circ}\right)\left(\left({ }^{\prime}\right) f\right)\right)\{P$. partition-on $\{0 . .<\operatorname{card} A\} P\}$
by (intro inj-on-impl-inj-on-image $[$ of - Pow $\{0 . .<$ card $A\}]$
inj-on-impl-inj-on-image[of - \{0..<card A\}]) blast+
moreover from inj have ( $)\left(\left({ }^{\prime}\right) f\right)^{\prime}\{P$. partition-on $\{0 . .<$ card $A\} P\}=\{P$. partition-on A $P\}$
by (subst image-f-eq, auto elim!: set-of-partition-on-map)
ultimately have bij-betw ((') ((`)f)) \{P. partition-on $\{0 . .<$ card $A\} P\}\{P$.
partition-on $A P\}$
by (auto intro: bij-betw-imageI)

```
    from this〈finite A〉 show ?thesis
    unfolding Bell-def
    by (subst bij-betw-iff-card[symmetric]) (auto intro: finitely-many-partition-on)
qed
lemma Bell-0:
    Bell 0 = 1
by (auto simp add: Bell-def partition-on-empty)
```


### 1.3 Construction of the Partitions

definition construct-partition-on $::$ 'a set $\Rightarrow{ }^{\prime} a$ set $\Rightarrow$ 'a set set set where
construct-partition-on $B C=$
do \{
$k \leftarrow\{0$..card B\};
$j \leftarrow\{0$..card $C\} ;$
$P \leftarrow\{P$. partition-on $C P \wedge$ card $P=j\}$;
$B^{\prime} \leftarrow\left\{B^{\prime} . B^{\prime} \subseteq B \wedge\right.$ card $\left.B^{\prime}=k\right\} ;$
$Q \leftarrow\left\{Q\right.$. partition-on $\left.B^{\prime} Q\right\} ;$
$f \leftarrow\left(B-B^{\prime}\right) \rightarrow_{E} P ;$
$P^{\prime} \leftarrow\left\{\left(\lambda X . X \cup\left\{x \in B-B^{\prime} . f x=X\right\}\right) \quad\right.$ ' $\left.P\right\} ;$
$\left\{P^{\prime} \cup Q\right\}$
\}
lemma construct-partition-on:
assumes finite $B$ finite $C$
assumes $B \cap C=\{ \}$
shows construct-partition-on $B C=\{P$. partition-on $(B \cup C) P\}$
proof (rule set-eqI')
fix $Q^{\prime}$
assume $Q^{\prime} \in$ construct-partition-on $B C$
from this obtain $j k P P^{\prime} Q B^{\prime} f$
where $j \leq \operatorname{card} C$
and $k \leq \operatorname{card} B$
and $P$ : partition-on $C P \wedge$ card $P=j$
and $B^{\prime}: B^{\prime} \subseteq B \wedge \operatorname{card} B^{\prime}=k$
and $Q$ : partition-on $B^{\prime} Q$
and $f: f \in B-B^{\prime} \rightarrow_{E} P$
and $P^{\prime}: P^{\prime}=\left(\lambda X . X \cup\left\{x \in B-B^{\prime} . f x=X\right\}\right)$ ' $P$
and $Q^{\prime}: Q^{\prime}=P^{\prime} \cup Q$
unfolding construct-partition-on-def by auto
from $P f$ have partition-on $\left(B-B^{\prime} \cup C\right) P^{\prime}$
unfolding $P^{\prime}$ using $\langle B \cap C=\{ \}\rangle$
by (intro partition-on-insert-elements) auto
from this $Q$ have partition-on $\left(\left(B-B^{\prime} \cup C\right) \cup B^{\prime}\right) Q^{\prime}$
unfolding $Q^{\prime}$ using $B^{\prime}\langle B \cap C=\{ \}\rangle$ by (auto intro: partition-on-union)
from this have partition-on $(B \cup C) Q^{\prime}$
using $B^{\prime}$ by (metis Diff-partition sup.assoc sup.commute)

```
    from this show }\mp@subsup{Q}{}{\prime}\in{P\mathrm{ . partition-on (BUC)P} by auto
next
    fix }\mp@subsup{Q}{}{\prime
    assume }\mp@subsup{Q}{}{\prime}:\mp@subsup{Q}{}{\prime}\in{\mp@subsup{Q}{}{\prime}.\mathrm{ partition-on }(B\cupC)\mp@subsup{Q}{}{\prime}
    from Q 采 have {} & Q' by (auto elim!: partition-onE)
    obtain Q where Q:Q = ((\lambdaX. if X\subseteqB then X else {})' Q') - {{}} by blast
    obtain }\mp@subsup{P}{}{\prime}\mathrm{ where }\mp@subsup{P}{}{\prime}:\mp@subsup{P}{}{\prime}=((\lambdaX\mathrm{ . if }X\subseteqB\mathrm{ then {} else X) ' }\mp@subsup{Q}{}{\prime})-{{}}\mathrm{ by
blast
    from }\mp@subsup{P}{}{\prime}Q<{}\not\in\mp@subsup{Q}{}{\prime}>\mathrm{ have }\mp@subsup{Q}{}{\prime}\mathrm{ -prop: }\mp@subsup{Q}{}{\prime}=\mp@subsup{P}{}{\prime}\cupQ\mathrm{ by auto
    have }\mp@subsup{P}{}{\prime}\mathrm{ -nosubset: }\forallX\in\mp@subsup{P}{}{\prime}.\negX\subseteq
        unfolding }\mp@subsup{P}{}{\prime}\mathrm{ by auto
    moreover have }\forallX\in\mp@subsup{P}{}{\prime}.X\subseteqB\cup
    using }\mp@subsup{Q}{}{\prime}\mp@subsup{P}{}{\prime}\mathrm{ by (auto elim: partition-onE)
    ultimately have }\mp@subsup{P}{}{\prime}\mathrm{ -witness: }\forallX\in\mp@subsup{P}{}{\prime}.\existsx.x\inX\cap
    using }\langleB\capC={}\rangle\mathrm{ by fastforce
    obtain }\mp@subsup{B}{}{\prime}\mathrm{ where }\mp@subsup{B}{}{\prime}:\mp@subsup{B}{}{\prime}=\bigcupQ by blas
    have Q-prop: partition-on }\mp@subsup{B}{}{\prime}
    using B' Q' Q'-prop partition-on-split2 mem-Collect-eq by blast
    have \cup}\cup\mp@subsup{P}{}{\prime}=B-\mp@subsup{B}{}{\prime}\cup
    proof
    have \bigcup Q ' = B\cupC\forallX\inQ'.\forall\mp@subsup{X}{}{\prime}\in\mp@subsup{Q}{}{\prime}.X\not=\mp@subsup{X}{}{\prime}\longrightarrowX\cap\mp@subsup{X}{}{\prime}={}
        using \mp@subsup{Q}{}{\prime}}\mathrm{ unfolding partition-on-def disjoint-def by auto
    from this show }\bigcup\mp@subsup{P}{}{\prime}\subseteqB-\mp@subsup{B}{}{\prime}\cup
        unfolding P' 政Q by auto blast
    next
    show }B-\mp@subsup{B}{}{\prime}\cupC\subseteq\bigcup\mp@subsup{P}{}{\prime
    proof
            fix }
            assume }x\inB-\mp@subsup{B}{}{\prime}\cup
            from this obtain X where X:x\inX X\inQ'
            using Q' by (metis Diff-iff Un-iff mem-Collect-eq partition-on-partition-on-unique)
            have }\forallX\in\mp@subsup{Q}{}{\prime}.X\subseteqB\longrightarrowX\subseteq\mp@subsup{B}{}{\prime
                unfolding \mp@subsup{B}{}{\prime}Q}\mathrm{ by auto
            from this X <x \inB - B'\cupC> have }\negX\subseteq
                using <B\capC={}> by auto
            from this }{X\in\mp@subsup{Q}{}{\prime}\rangle\mathrm{ have }X\in\mp@subsup{P}{}{\prime}\mathrm{ using }\mp@subsup{P}{}{\prime}\mathrm{ by auto
            from this }\langlex\inX\rangle\mathrm{ show }x\in\bigcup\mp@subsup{P}{}{\prime}\mathrm{ by auto
        qed
    qed
    from this have partition-on-P': partition-on (B-\mp@subsup{B}{}{\prime}\cupC) P'
        using partition-on-split1 ('-prop Q' mem-Collect-eq by fastforce
    obtain P where P: P}=(\lambdaX.X\capC)' P' by blas
    from P partition-on-P' P'-witness have partition-on C P
        using partition-on-intersect-on-elements by auto
    obtain f where f:f=( }\lambdax\mathrm{ . if }x\inB-\mp@subsup{B}{}{\prime}\mathrm{ then (THE X. x }\inX\wedgeX\in\mp@subsup{P}{}{\prime})
C else undefined) by blast
    have }\mp@subsup{P}{}{\prime}\mathrm{ -prop: }\mp@subsup{P}{}{\prime}=(\lambdaX.X\cup{x\inB-\mp@subsup{B}{}{\prime}.fx=X})'
    proof
    {
```

```
    fix }
    assume }X\in\mp@subsup{P}{}{\prime
    have X-subset: }X\subseteq(B-\mp@subsup{B}{}{\prime})\cup
    using partition-on-P}\mp@subsup{P}{}{\prime}\langleX\in\mp@subsup{P}{}{\prime}\rangle\mathrm{ by (auto elim: partition-onE)
    have }X=X\capC\cup{x\inB-\mp@subsup{B}{}{\prime}.fx=X\capC
    proof
    {
        fix }
        assume }x\in
        from this X-subset have }x\in(B-\mp@subsup{B}{}{\prime})\cupC\mathrm{ by auto
        from this have }x\inX\capC\cup{xa\inB-\mp@subsup{B}{}{\prime}.fxa=X\capC
        proof
            assume }x\in
            from this }\langlex\inX\rangle\mathrm{ show ?thesis by simp
        next
            assume }x\inB-\mp@subsup{B}{}{\prime
```



```
P')}=
            by (simp add: partition-on-the-part-eq)
            from }<x\inB-\mp@subsup{B}{}{\prime}\rangle\mathrm{ this show ?thesis unfolding }f\mathrm{ by auto
        qed
    }
    from this show }X\subseteqX\capC\cup{x\inB-\mp@subsup{B}{}{\prime}.fx=X\capC} by aut
next
    show }X\capC\cup{xa\inB-\mp@subsup{B}{}{\prime}.fxa=X\capC}\subseteq
    proof
        fix }
        assume }x\inX\capC\cup{x\inB-\mp@subsup{B}{}{\prime}.fx=X\capC
        from this show }x\in
        proof
            assume }x\inX\cap
            from this show ?thesis by simp
    next
        assume x-in: }x\in{x\inB-\mp@subsup{B}{}{\prime}.fx=X\capC
        from this have ex1: }\exists!X.X.x\inX\wedgeX\in\mp@subsup{P}{}{\prime
        using partition-on-P' by (auto intro!: partition-on-partition-on-unique)
        from x-in X-subset have eq:(THE X. x 位^X\in P') 
            unfolding }f\mathrm{ by auto
        from P'-nosubset }\langleX\in\mp@subsup{P}{}{\prime}\rangle\mathrm{ have }\negX\subseteqB\mathrm{ by simp
        from this have }X\capC\not={
            using X-subset assms(3) by blast
        from this obtain }y\mathrm{ where y: y }\inX\capC\mathrm{ by auto
```



```
        from y y-in have }y\inXy\in(THEX.x\inX\wedgeX\in\mp@subsup{P}{}{\prime})\mathrm{ by auto
        moreover from y have }\exists!X.y\inX\wedgeX\in\mp@subsup{P}{}{\prime
            using partition-on-P' by (simp add: partition-on-partition-on-unique)
        moreover have (THE X. x X X \ X \in P') \in P'
            using ex1 by (rule the1I2) auto
```


auto
qed
qed
qed
from $\left\langle X \in P^{\prime}\right\rangle$ this have $X \in\left(\lambda X . X \cup\left\{x \in B-B^{\prime} . f x=X\right\}\right)$ ' $P$
unfolding $P$ by simp
\}
from this show $P^{\prime} \subseteq\left(\lambda X . X \cup\left\{x \in B-B^{\prime} . f x=X\right\}\right)$ ' $P .$.
next
\{
fix $x$
assume $x$-in-image: $x \in\left(\lambda X . X \cup\left\{x \in B-B^{\prime} . f x=X\right\}\right)$ ' $P$
\{
fix $X$
assume $X \in P^{\prime}$
have $\left\{x \in B-B^{\prime} . f x=X \cap C\right\}=\left\{x \in B-B^{\prime} . x \in X\right\}$
proof -
\{
fix $x$
assume $x \in B-B^{\prime}$
from this have ex1: $\exists!X . x \in X \wedge X \in P^{\prime}$
using partition-on- $P^{\prime}$ by (auto intro!: partition-on-partition-on-unique)
from this have in-p: $\left(T H E X . x \in X \wedge X \in P^{\prime}\right) \in P^{\prime}$
and $x$-in: $x \in\left(T H E X . x \in X \wedge X \in P^{\prime}\right)$
by (metis (mono-tags, lifting) theI) +
have $f x=X \cap C \longleftrightarrow\left(T H E X . x \in X \wedge X \in P^{\prime}\right) \cap C=X \cap C$
using $\left\langle x \in B-B^{\prime}\right\rangle$ unfolding $f$ by auto
also have $\ldots \longleftrightarrow\left(\right.$ THE $\left.X . x \in X \wedge X \in P^{\prime}\right)=X$
proof
assume (THE $X . x \in X \wedge X \in P^{\prime}$ ) $=X$
from this show (THE $\left.X . x \in X \wedge X \in P^{\prime}\right) \cap C=X \cap C$ by auto
next
assume (THE $\left.X . x \in X \wedge X \in P^{\prime}\right) \cap C=X \cap C$
have (THE $\left.X . x \in X \wedge X \in P^{\prime}\right) \cap X \neq\{ \}$
using $P^{\prime}$-witness $\left\langle\left(T H E X . x \in X \wedge X \in P^{\prime}\right) \cap C=X \cap C\right\rangle\langle X \in$
$P^{\prime}>$ by fastforce
from this show (THE $\left.X . x \in X \wedge X \in P^{\prime}\right)=X$
using partition-on- $P^{\prime}[$ unfolded partition-on-def disjoint-def] in-p $\langle X$
$\in P^{\prime}>$ by metis
qed
also have $\ldots \longleftrightarrow x \in X$
using ex1 $\left\langle X \in P^{\prime}\right\rangle x$-in by (auto; metis (no-types, lifting) the-equality)
finally have $f x=X \cap C \longleftrightarrow x \in X$.
\}
from this show ?thesis by auto
qed
moreover have $X \subseteq B-B^{\prime} \cup C$
using partition-on- $P^{\prime}\left\langle X \in P^{\prime}\right\rangle$ by (blast elim: partition-onE)
ultimately have $X \cap C \cup\left\{x \in B . x \notin B^{\prime} \wedge f x=X \cap C\right\}=X$ by auto \}
from this $x$-in-image have $x \in P^{\prime}$ unfolding $P$ by auto
\}
from this show $\left(\lambda X . X \cup\left\{x \in B-B^{\prime} . f x=X\right\}\right)$ ' $P \subseteq P^{\prime} .$.
qed
from partition-on- $P^{\prime}$ have f-prop: $f \in\left(B-B^{\prime}\right) \rightarrow_{E} P$
unfolding $f P$ by (auto simp add: partition-on-the-part-mem)
from $Q B^{\prime}$ have $B^{\prime} \subseteq B$ by auto
obtain $k$ where $k: k=\operatorname{card} B^{\prime}$ by blast
from $\langle$ finite $B\rangle\left\langle B^{\prime} \subseteq B\right\rangle k$ have $k$-prop: $k \in\{0$..card $B\}$ by (simp add: card-mono)
obtain $j$ where $j: j=$ card $P$ by blast
from $j$ ppartition-on $C P\rangle$ have $j$-prop: $j \in\{0$..card $C\}$
by (simp add: assms(2) partition-on-le-set-elements)
from 〈partition-on $C P>j$ have $P$-prop: partition-on $C P \wedge$ card $P=j$ by auto
from $k\left\langle B^{\prime} \subseteq B\right\rangle$ have $B^{\prime}$-prop: $B^{\prime} \subseteq B \wedge$ card $B^{\prime}=k$ by auto
show $Q^{\prime} \in$ construct-partition-on $B C$
using j-prop $k$-prop $P$-prop $B^{\prime}$-prop $Q$-prop $P^{\prime}$-prop $f$-prop $Q^{\prime}$-prop
unfolding construct-partition-on-def
by (auto simp del: atLeastAtMost-iff) blast
qed

### 1.4 Injectivity of the Set Construction

lemma injectivity:
assumes $B \cap C=\{ \}$
assumes $P:($ partition-on $C P \wedge \operatorname{card} P=j) \wedge\left(\right.$ partition-on $C P^{\prime} \wedge$ card $P^{\prime}=$ $j^{\prime}$ )
assumes $B^{\prime}:\left(B^{\prime} \subseteq B \wedge \operatorname{card} B^{\prime}=k\right) \wedge\left(B^{\prime \prime} \subseteq B \wedge \operatorname{card} B^{\prime \prime}=k^{\prime}\right)$
assumes $Q$ : partition-on $B^{\prime} Q \wedge$ partition-on $B^{\prime \prime} Q^{\prime}$
assumes $f: f \in B-B^{\prime} \rightarrow_{E} P \wedge g \in B-B^{\prime \prime} \rightarrow_{E} P^{\prime}$
assumes $P^{\prime}: P^{\prime \prime}=\left(\lambda X . X \cup\left\{x \in B-B^{\prime} . f x=X\right\}\right)$ ' $P \wedge$
$P^{\prime \prime \prime}=\left(\lambda X . X \cup\left\{x \in B-B^{\prime \prime} . g x=X\right\}\right)$ ' $P^{\prime}$
assumes eq-result: $P^{\prime \prime} \cup Q=P^{\prime \prime \prime} \cup Q^{\prime}$
shows $f=g$ and $Q=Q^{\prime}$ and $B^{\prime}=B^{\prime \prime}$
and $P=P^{\prime}$ and $j=j^{\prime}$ and $k=k^{\prime}$
proof -
have $P$-nonempty-sets: $\forall X \in P . \exists c \in C . c \in X \forall X \in P^{\prime} . \exists c \in C . c \in X$
using $P$ by (force elim: partition-onE)+
have 1: $\forall X \in P^{\prime \prime} . \exists c \in C . c \in X \forall X \in P^{\prime \prime \prime} . \exists c \in C . c \in X$ using $P^{\prime} P$-nonempty-sets by fastforce +
have 2: $\forall X \in Q . \forall x \in X . x \notin C \forall X \in Q^{\prime} . \forall x \in X . x \notin C$
using $\langle B \cap C=\{ \}\rangle Q B^{\prime}$ by (auto elim: partition-onE)
from eq-result have $P^{\prime \prime}=P^{\prime \prime \prime}$ and $Q=Q^{\prime}$
by (auto dest: injectivity-union[OF - 1 2])
from this $Q$ show $Q=Q^{\prime}$ and $B^{\prime}=B^{\prime \prime}$
by (auto intro!: partition-on-eq-implies-eq-carrier)
have subset- $C: \forall X \in P . X \subseteq C \forall X \in P^{\prime} . X \subseteq C$
using $P$ by (auto elim: partition-onE)
have eq-image: $\left(\lambda X . X \cup\left\{x \in B-B^{\prime} . f x=X\right\}\right)^{\prime} P=(\lambda X . X \cup\{x \in B-$ $\left.\left.B^{\prime \prime} . g x=X\right\}\right)$ ' $P^{\prime}$
using $\left.P^{\prime}{ }^{\langle } P^{\prime \prime}=P^{\prime \prime \prime}\right\rangle$ by auto
from this $\langle B \cap C=\{ \}\rangle$ show $P=P^{\prime}$
by (auto dest: injectivity-image-union $[O F-$ subset- $C]$ )
have eq2: $\left(\lambda X . X \cup\left\{x \in B-B^{\prime} . f x=X\right\}\right)$ ' $P=\left(\lambda X . X \cup\left\{x \in B-B^{\prime} . g\right.\right.$ $x=X\})$ ' $P$
using $\left\langle P=P^{\prime}\right\rangle\left\langle B^{\prime}=B^{\prime \prime}\right\rangle$ eq-image by simp
from $P$ have $P$-props: $\forall X \in P . X \subseteq C \forall X \in P . X \neq\{ \}$ by (auto elim: partition-onE)
have invert: $\forall X \in P .\left(X \cup\left\{x \in B-B^{\prime} . f x=X\right\}\right) \cap C=X \wedge(X \cup\{x \in B$ $\left.\left.-B^{\prime} . g x=X\right\}\right) \cap C=X$
using $\langle B \cap C=\{ \}\rangle P$-props by auto
have eq3: $\forall X \in P .\left(X \cup\left\{x \in B-B^{\prime} . f x=X\right\}\right)=\left(X \cup\left\{x \in B-B^{\prime} . g x=\right.\right.$ $X\}$ )
using injectivity-image[OF eq2 invert] by blast
have eq4: $\forall X \in P .\left\{x \in B-B^{\prime} . f x=X\right\}=\left\{x \in B-B^{\prime} . g x=X\right\}$
proof
fix $X$
assume $X \in P$
from this $P$ have $X \subseteq C$ by (auto elim: partition-onE)
have disjoint: $X \cap\left\{x \in B-B^{\prime} . f x=X\right\}=\{ \} X \cap\left\{x \in B-B^{\prime} . g x=X\right\}$ $=\{ \}$
using $\langle B \cap C=\{ \}\rangle\langle X \subseteq C\rangle$ by auto
from eq $3\langle X \in P\rangle$ have $X \cup\left\{x \in B-B^{\prime} . f x=X\right\}=X \cup\left\{x \in B-B^{\prime} . g\right.$ $x=X\}$ by auto
from this disjoint show $\left\{x \in B-B^{\prime} . f x=X\right\}=\left\{x \in B-B^{\prime} . g x=X\right\}$
by (auto intro: injectivity-union)
qed
from eq4 $f$ have eq5: $\forall b \in B-B^{\prime} . f b=g b$ by blast
from eq5 $f\left\langle B^{\prime}=B^{\prime \prime}\right\rangle\left\langle P=P^{\prime}\right\rangle$ show eq6: $f=g$ by (auto intro: exten-sional-funcset-ext)
from $P\left\langle P=P^{\prime}\right\rangle$ show $j=j^{\prime}$ by simp
from $B^{\prime}\left\langle B^{\prime}=B^{\prime \prime}\right\rangle$ show $k=k^{\prime}$ by simp
qed

### 1.5 The Generalized Bell Recurrence Relation

theorem Bell-eq:
Bell $(n+m)=\left(\sum k \leq n . \sum j \leq m . j^{\wedge}(n-k) *\right.$ Stirling $m j *(n$ choose $k) *$ Bell k)
proof -
define $A$ where $A=\{0 . .<n+m\}$
define $B$ where $B=\{0 . .<n\}$
define $C$ where $C=\{n . .<n+m\}$
have $A=B \cup C B \cap C=\{ \}$ finite $B$ card $B=n$ finite $C$ card $C=m$
unfolding $A$-def $B$-def $C$-def by auto
have step 1: Bell $(n+m)=\operatorname{card}\{P$. partition-on $A P\}$
unfolding Bell－def $A$－def ．．
from $\langle A=B \cup C\rangle\langle B \cap C=\{ \}\rangle\langle$ finite $B\rangle\langle$ finite $C\rangle$
have step2：card $\{P$ ．partition－on $A P\}=$ card（construct－partition－on $B C$ ）
by（simp add：construct－partition－on）
note injectivity $=$ injectivity $[O F\langle B \cap C=\{ \}\rangle]$
let ？expr $=d o\{$
$k \leftarrow\{0 .$. card $B\} ;$
$j \leftarrow\{0$ ．．card $C\} ;$
$P \leftarrow\{P$ ．partition－on $C P \wedge$ card $P=j\} ;$
$B^{\prime} \leftarrow\left\{B^{\prime} . B^{\prime} \subseteq B \wedge\right.$ card $\left.B^{\prime}=k\right\} ;$
$Q \leftarrow\left\{Q\right.$ ．partition－on $\left.B^{\prime} Q\right\} ;$
$f \leftarrow\left(B-B^{\prime}\right) \rightarrow_{E} P ;$
$P^{\prime} \leftarrow\left\{\left(\lambda X . X \cup\left\{x \in B-B^{\prime} . f x=X\right\}\right)^{\prime} P\right\} ;$
$\left\{P^{\prime} \cup Q\right\}$
\}
let ？S $\gg$ ？$c o m p=$ ？ $\operatorname{expr}$
\｛
fix $k$
assume $k: k \in\{.$. card $B\}$
let ？expr $=$ ？comp $k$
let ？$S \gg$ ？ comp $=$ ？expr
\｛
fix $j$
assume $j \in\{$ ．．card $C\}$
let ？expr $=$ ？comp $j$
let ？$S \gg$ ？$c o m p=$ ？expr
from〈finite $C$ 〉 have finite？$S$
by（intro finite－Collect－conjI disjI1 finitely－many－partition－on）
\｛
fix $P$
assume $P: P \in\{P$ ．partition－on $C P \wedge$ card $P=j\}$
from this have partition－on C $P$ by simp
let ？expr＝？comp $P$
let ？$S \gg$ ？$c o m p=$ ？expr
have finite $P$
using $P$ 〈finite $C$ 〉 by（auto intro：finite－elements）
from 〈finite $B$ 〉have finite ？$S$ by（auto simp add：finite－subset）
moreover
\｛
fix $B^{\prime}$
assume $B^{\prime}: B^{\prime} \in\left\{B^{\prime} . B^{\prime} \subseteq B \wedge\right.$ card $\left.B^{\prime}=k\right\}$
from this have $B^{\prime} \subseteq B$ by simp
let ？expr $=$ ？comp $B^{\prime}$
let ？$S \gg$ ？comp＝？expr
from $\langle$ finite $B\rangle$ have finite $B^{\prime}$
using $B^{\prime}$ by（auto simp add：finite－subset）
from 〈finite $\left.B^{\prime}\right\rangle$ have finite $\left\{Q\right.$ ．partition－on $\left.B^{\prime} Q\right\}$
by（rule finitely－many－partition－on）
moreover

```
{
    fix }
    assume Q:Q Q {Q. partition-on B' Q}
    let ?expr = ?comp Q
    let ?S>> ?comp = ? expr
    {
        fix f
        assume f\inB-B'㕵 P
        let ?expr = ?comp f
        let ?S >> ?comp = ?expr
        have disjoint-family-on ?comp ?S
        by (auto intro: disjoint-family-onI)
        from this have card ?expr = 1
        by (simp add: card-bind-constant)
    moreover have finite?expr
        by (simp add: finite-bind)
    ultimately have finite ?expr ^ card ?expr = 1 by blast
    }
    moreover have finite?S
        using 〈finite B\rangle<finite P> by (auto intro: finite-PiE)
    moreover have disjoint-family-on ?comp ?S
    using P B' Q
    by (injectivity-solver rule: local.injectivity(1))
    moreover have card ?S = j^(n-k)
    proof -
    have card (B-B) =n-k
        using B'<finite B'\ <card B = n>
        by (subst card-Diff-subset) auto
    from this show ?thesis
        using \finite B\rangleP
        by (subst card-PiE) (simp add: prod-constant)+
    qed
    ultimately have card ? expr = j^(n-k)
        by (simp add: card-bind-constant)
    moreover have finite? expr
    using \finite? ?S`\finite {P.partition-on C P\wedge card P=j}>
    by (auto intro!: finite-bind)
    ultimately have finite ?expr ^ card ?expr = j^(n-k) by blast
} note inner = this
moreover have card ?S = Bell k
    using B'^finite B'> by (auto simp add: Bell-altdef[symmetric])
moreover have disjoint-family-on ?comp ?S
    using P B'
    by (injectivity-solver rule: local.injectivity(2))
ultimately have card ? expr = j^(n-k)* Bell k
    by (subst card-bind-constant) auto
moreover have finite? expr
    using inner <finite ?S` by (auto intro: finite-bind)
ultimately have finite ?expr ^card ?expr = j^ ( }n-k)*\mathrm{ Bell }k\mathrm{ by blast
```

```
    } note inner = this
    moreover have card ?S = n choose k
        using<card B=n\rangle<finite B> by (simp add: n-subsets)
    moreover have disjoint-family-on ?comp?S
        using P
        by (injectivity-solver rule: local.injectivity(3))
    ultimately have card ? expr = j^ ( n-k)* (n choose k)* Bell k
        by (subst card-bind-constant) auto
        moreover have finite? expr
            using inner <finite ?S` by (auto intro: finite-bind)
        ultimately have finite ? expr ^ card ? expr = j^(n-k)* (n choose k)*
Bell k by blast
    } note inner = this
    moreover note〈finite?S>
    moreover have card ?S = Stirling m j
        using <finite C\rangle\langlecard C=m\rangle by (simp add:card-partition-on)
    moreover have disjoint-family-on ?comp ?S
        by (injectivity-solver rule: local.injectivity(4))
        ultimately have card ? expr = j^(n-k)*Stirling m j* (n choose k)*
Bell k
            by (subst card-bind-constant) auto
    moreover have finite? expr
            using inner <finite ?S> by (auto intro: finite-bind)
    ultimately have finite ? expr ^ card ?expr = j^(n-k)*Stirling m j* (n
choose k) * Bell k by blast
    } note inner = this
    moreover have finite?S by simp
    moreover have disjoint-family-on ?comp ?S
        by (injectivity-solver rule: local.injectivity(5))
    ultimately have card ? expr = (\sumj\leqm. j^}(n-k)*Stirling m j* (n choose
k)* Bell k) (is - = ?formula)
            using <card C = m> by (subst card-bind) (auto intro: sum.cong)
            moreover have finite ? expr
            using inner 〈finite ?S> by (auto intro: finite-bind)
    ultimately have finite ?expr ^ card ? expr = ?formula by blast
}
    moreover have finite ?S by simp
    moreover have disjoint-family-on ?comp ?S
    by (injectivity-solver rule: local.injectivity(6))
    ultimately have step3: card (construct-partition-on B C) = (\sumk\leqn. \sumj\leqm.
j^(n-k)*Stirling mj* (n choose k)* Bell k)
    unfolding construct-partition-on-def
    using <card B = n〉 by (subst card-bind) (auto intro: sum.cong)
    from step1 step2 step3 show ?thesis by auto
qed
```


## 1．6 Corollaries of the Generalized Bell Recurrence

corollary Bell－Stirling－eq：

```
    Bell m}=(\sumj\leqm.Stirling m j)
proof -
    have Bell m=Bell (0+m) by simp
    also have ... = (\sumj\leqm. Stirling m j)
        unfolding Bell-eq[of 0] by (simp add: Bell-0)
    finally show ?thesis.
qed
corollary Bell-recursive-eq:
    Bell (n+1) = (\sumk\leqn. (n choose k)* Bell k)
unfolding Bell-eq[of - 1] by simp
```


### 1.7 Code equations for the computation of Bell numbers

It is slow to compute Bell numbers without dynamic programming (DP). The following is a DP algorithm derived from the previous recursion formula Bell-recursive-eq.
fun Bell-list-aux :: nat $\Rightarrow$ nat list
where
Bell-list-aux $0=[1] \mid$
Bell-list-aux (Suc n) $=($
let prev-list $=$ Bell-list-aux n;
next-val $=\left(\sum(k, z) \leftarrow\right.$ List.enumerate 0 prev-list. $z *(n$ choose $\left.(n-k))\right)$
in next-val\#prev-list)
definition Bell-list $::$ nat $\Rightarrow$ nat list
where Bell-list $n=$ rev (Bell-list-aux $n$ )
lemma bell-list-eq: Bell-list $n=$ map Bell $[0 . .<n+1]$
proof -
have Bell-list-aux $n=\operatorname{rev}($ map Bell $[0 . .<$ Suc $n])$
proof (induction n)
case 0
then show ?case by (simp add:Bell-0)
next
case (Suc n)
define $x$ where $x=$ Bell-list-aux $n$
define $y$ where $y=\left(\sum(k, z) \leftarrow\right.$ List.enumerate $0 x . z *(n$ choose $\left.(n-k))\right)$
define $s n$ where $s n=n+1$
have $b: x=\operatorname{rev}($ map Bell $[0 . .<s n])$
using Suc $x$-def sn-def by simp
have $c$ : length $x=s n$
unfolding $b$ by simp
have snd $i=\operatorname{Bell}(n-f s t i)$ if $i \in \operatorname{set}($ List.enumerate $0 x)$ for $i$
proof -
have fst $i<$ length $x$ snd $i=x!$ fst $i$
using iffD 1 [OF in-set-enumerate-eq that $]$ by auto
hence snd $i=\operatorname{Bell}(s n-S u c(f s t i))$

```
            unfolding b by (simp add:rev-nth)
            thus ?thesis
            unfolding sn-def by simp
    qed
    hence }y=(\sumi\leftarrow\mathrm{ enumerate 0x. Bell ( }n-fst i)*(n\mathrm{ choose ( }n-fst i))
        unfolding }y\mathrm{ -def by (intro arg-cong[where f=sum-list] map-cong refl)
            (simp add:case-prod-beta)
    also have ... = (\sumi\leftarrowmap fst (enumerate 0 x). Bell ( }n-i)*(n\mathrm{ choose ( }n
i)))
            by (subst map-map) (simp add:comp-def)
    also have ... = (\sumi=0..<length x. Bell ( }n-i)*(n\mathrm{ choose ( }n-i))
        by (simp add:interv-sum-list-conv-sum-set-nat)
    also have ... = (\sumi\leqn. Bell (n-i)*(n choose ( }n-i))
    using c sn-def by (intro sum.cong) auto
    also have ... = (\sumi\in(\lambdak.n-k)'{..n}. Bell i*(n choose i))
        by (subst sum.reindex, auto simp add:inj-on-def)
    also have ... = (\sumi\leqn. Bell i*( }n\mathrm{ choose i)}
        by (intro sum.cong refl iffD2[OF set-eq-iff] allI)
            (simp add:image-iff atMost-def, presburger)
    also have ... = Bell (Suc n)
        using Bell-recursive-eq by (simp add:mult.commute)
    finally have a:y = Bell (Suc n) by simp
    have Bell-list-aux (Suc n)=y#x
        unfolding x-def y-def by (simp add:Let-def)
    also have ... = Bell (Suc n)#(rev (map Bell [0..<Suc n]))
        unfolding a b sn-def by simp
    also have ... = rev (map Bell [0..<Suc (Suc n)])
        by simp
    finally show ?case by simp
qed
thus Bell-list n = map Bell [0..<n+1]
    by (simp add:Bell-list-def)
qed
lemma Bell-eval[code]: Bell n = last (Bell-list n)
    unfolding bell-list-eq by simp
end
```


## References

[1] N. J. A. Sloane. A000110: Bell or exponential numbers: number of ways to partition a set of n labeled elements. In The On-Line Encyclopedia of Integer Sequences. https://oeis.org/A000110.
[2] M. Z. Spivey. A generalized recurrence for Bell numbers. Journal of

Integer Sequences, 11, 2008. Electronic copy available at https://cs.uwaterloo.ca/journals/JIS/VOL11/Spivey/spivey25.pdf.

