

# Matroids

Jonas Keinholz

June 6, 2026

## **Abstract**

This article defines combinatorial structures known as *Independence Systems* and *Matroids* and provides basic concepts and theorems related to them. These structures play an important role in combinatorial optimisation, e. g. greedy algorithms such as Kruskal's algorithm. The development is based on Oxley's 'What is a Matroid?' [1].

# Contents

<b>1</b>	<b>Independence systems</b>	<b>3</b>
1.1	Sub-independence systems . . . . .	3
1.2	Bases . . . . .	5
1.3	Circuits . . . . .	8
1.4	Relation between independence and bases . . . . .	12
1.5	Relation between dependence and circuits . . . . .	14
1.6	Ranks . . . . .	15
<b>2</b>	<b>Matroids</b>	<b>17</b>
2.1	Minors . . . . .	18
2.2	Bases . . . . .	19
2.3	Circuits . . . . .	20
2.4	Ranks . . . . .	26
2.5	Closure . . . . .	33

# 1 Independence systems

```
theory Indep-System
  imports Main
begin
```

```
lemma finite-psubset-inc-induct:
  assumes finite A X  $X \subseteq A$ 
  assumes  $\bigwedge X. (\bigwedge Y. X \subset Y \implies Y \subseteq A \implies P Y) \implies P X$ 
  shows P X
proof -
  have wf: wf  $\{(X, Y). Y \subset X \wedge X \subseteq A\}$ 
  by (rule wf-bounded-set[where ub =  $\lambda-. A$  and f = id]) (auto simp add:  $\langle$ finite A $\rangle$ )
  show ?thesis
  proof (induction X rule: wf-induct[OF wf, case-names step])
    case (step X)
    then show ?case using assms(3)[of X] by blast
  qed
qed
```

An *independence system* consists of a finite ground set together with an independence predicate over the sets of this ground set. At least one set of the carrier is independent and subsets of independent sets are also independent.

```
locale indep-system =
  fixes carrier :: 'a set
  fixes indep :: 'a set  $\implies$  bool
  assumes carrier-finite: finite carrier
  assumes indep-subset-carrier: indep X  $\implies X \subseteq$  carrier
  assumes indep-ex:  $\exists X. indep X$ 
  assumes indep-subset: indep X  $\implies Y \subseteq X \implies indep Y$ 
begin
```

```
lemmas psubset-inc-induct [case-names carrier step] = finite-psubset-inc-induct[OF carrier-finite]
```

```
lemmas indep-finite [simp] = finite-subset[OF indep-subset-carrier carrier-finite]
```

The empty set is independent.

```
lemma indep-empty [simp]: indep  $\{\}$ 
  using indep-ex indep-subset by auto
```

## 1.1 Sub-independence systems

A subset of the ground set induces an independence system.

```
definition indep-in where indep-in  $\mathcal{E} X \longleftrightarrow X \subseteq \mathcal{E} \wedge indep X$ 
```

```
lemma indep-inI:
  assumes  $X \subseteq \mathcal{E}$ 
```

**assumes** *indep*  $X$   
**shows** *indep-in*  $\mathcal{E} X$   
**using** *assms unfolding indep-in-def* **by** *auto*

**lemma** *indep-in-subI*: *indep-in*  $\mathcal{E} X \implies \text{indep-in } \mathcal{E}' (X \cap \mathcal{E}')$   
**using** *indep-subset unfolding indep-in-def* **by** *auto*

**lemma** *dep-in-subI*:  
**assumes**  $X \subseteq \mathcal{E}'$   
**shows**  $\neg \text{indep-in } \mathcal{E}' X \implies \neg \text{indep-in } \mathcal{E} X$   
**using** *assms unfolding indep-in-def* **by** *auto*

**lemma** *indep-in-subset-carrier*: *indep-in*  $\mathcal{E} X \implies X \subseteq \mathcal{E}$   
**unfolding** *indep-in-def* **by** *auto*

**lemma** *indep-in-subI-subset*:  
**assumes**  $\mathcal{E}' \subseteq \mathcal{E}$   
**assumes** *indep-in*  $\mathcal{E}' X$   
**shows** *indep-in*  $\mathcal{E} X$   
**proof** –  
**have** *indep-in*  $\mathcal{E} (X \cap \mathcal{E})$  **using** *assms indep-in-subI* **by** *auto*  
**moreover** **have**  $X \cap \mathcal{E} = X$  **using** *assms indep-in-subset-carrier* **by** *auto*  
**ultimately show** *?thesis* **by** *auto*  
**qed**

**lemma** *indep-in-supI*:  
**assumes**  $X \subseteq \mathcal{E}' \mathcal{E}' \subseteq \mathcal{E}$   
**assumes** *indep-in*  $\mathcal{E} X$   
**shows** *indep-in*  $\mathcal{E}' X$   
**proof** –  
**have**  $X \cap \mathcal{E}' = X$  **using** *assms* **by** *auto*  
**then show** *?thesis* **using** *assms indep-in-subI* [**where**  $\mathcal{E} = \mathcal{E}$  **and**  $\mathcal{E}' = \mathcal{E}'$  **and**  
 $X = X$ ] **by** *auto*  
**qed**

**lemma** *indep-in-indep*: *indep-in*  $\mathcal{E} X \implies \text{indep } X$   
**unfolding** *indep-in-def* **by** *auto*

**lemmas** *indep-inD = indep-in-subset-carrier indep-in-indep*

**lemma** *indep-system-subset* [*simp, intro*]:  
**assumes**  $\mathcal{E} \subseteq \text{carrier}$   
**shows** *indep-system*  $\mathcal{E}$  (*indep-in*  $\mathcal{E}$ )  
**unfolding** *indep-system-def indep-in-def*  
**using** *finite-subset[OF assms carrier-finite]* *indep-subset* **by** *auto*

We will work a lot with different sub structures. Therefore, every definition ‘foo’ will have a counterpart ‘foo\_in’ which has the ground set as an additional parameter. Furthermore, every result about ‘foo’ will have an-

other result about ‘foo\_in’. With this, we usually don’t have to work with **interpretation** in proofs.

**context**

fixes  $\mathcal{E}$

assumes  $\mathcal{E} \subseteq \text{carrier}$

**begin**

**interpretation**  $\mathcal{E}$ : *indep-system*  $\mathcal{E}$  *indep-in*  $\mathcal{E}$

using  $\langle \mathcal{E} \subseteq \text{carrier} \rangle$  **by** *auto*

**lemma** *indep-in-sub-cong*:

assumes  $\mathcal{E}' \subseteq \mathcal{E}$

shows  $\mathcal{E}.\text{indep-in } \mathcal{E}' X \longleftrightarrow \text{indep-in } \mathcal{E}' X$

**unfolding**  $\mathcal{E}.\text{indep-in-def}$  *indep-in-def* **using** *assms* **by** *auto*

**lemmas** *indep-in-ex* =  $\mathcal{E}.\text{indep-ex}$

**lemmas** *indep-in-subset* =  $\mathcal{E}.\text{indep-subset}$

**lemmas** *indep-in-empty* =  $\mathcal{E}.\text{indep-empty}$

**end**

## 1.2 Bases

A *basis* is a maximal independent set, i. e. an independent set which becomes dependent on inserting any element of the ground set.

**definition** *basis* **where**  $\text{basis } X \longleftrightarrow \text{indep } X \wedge (\forall x \in \text{carrier} - X. \neg \text{indep } (\text{insert } x X))$

**lemma** *basisI*:

assumes *indep*  $X$

assumes  $\bigwedge x. x \in \text{carrier} - X \implies \neg \text{indep } (\text{insert } x X)$

shows *basis*  $X$

**using** *assms* **unfolding** *basis-def* **by** *auto*

**lemma** *basis-indep*:  $\text{basis } X \implies \text{indep } X$

**unfolding** *basis-def* **by** *auto*

**lemma** *basis-max-indep*:  $\text{basis } X \implies x \in \text{carrier} - X \implies \neg \text{indep } (\text{insert } x X)$

**unfolding** *basis-def* **by** *auto*

**lemmas** *basisD* = *basis-indep* *basis-max-indep*

**lemmas** *basis-subset-carrier* = *indep-subset-carrier*[*OF* *basis-indep*]

**lemmas** *basis-finite* [*simp*] = *indep-finite*[*OF* *basis-indep*]

**lemma** *indep-not-basis*:

assumes *indep*  $X$

assumes  $\neg \text{basis } X$

shows  $\exists x \in \text{carrier} - X. \text{indep } (\text{insert } x X)$

```

using assms basisI by auto

lemma basis-subset-eq:
  assumes basis B1
  assumes basis B2
  assumes  $B_1 \subseteq B_2$ 
  shows  $B_1 = B_2$ 
proof (rule ccontr)
  assume  $B_1 \neq B_2$ 
  then obtain x where  $x \in B_2 - B_1$  using assms by auto
  then have  $\text{insert } x B_1 \subseteq B_2$  using assms by auto
  then have indep ( $\text{insert } x B_1$ ) using assms basis-indep[of  $B_2$ ] indep-subset by
auto
  moreover have  $x \in \text{carrier} - B_1$  using assms x basis-subset-carrier by auto
  ultimately show False using assms basisD by auto
qed

definition basis-in where
  basis-in  $\mathcal{E} X \longleftrightarrow \text{indep-system.basis } \mathcal{E} (\text{indep-in } \mathcal{E}) X$ 

lemma basis-iff-basis-in: basis B  $\longleftrightarrow$  basis-in carrier B
proof -
  interpret  $\mathcal{E}$ : indep-system carrier indep-in carrier
  by auto

  show basis B  $\longleftrightarrow$  basis-in carrier B
  unfolding basis-in-def
  proof (standard, goal-cases LTR RTL)
  case LTR
  show ?case
  proof (rule  $\mathcal{E}.\text{basisI}$ )
  show indep-in carrier B using LTR basisD indep-subset-carrier indep-inI by
auto
  next
  fix x
  assume  $x \in \text{carrier} - B$ 
  then have  $\neg \text{indep} (\text{insert } x B)$  using LTR basisD by auto
  then show  $\neg \text{indep-in carrier} (\text{insert } x B)$  using indep-inD by auto
  qed
  next
  case RTL
  show ?case
  proof (rule basisI)
  show indep B using RTL  $\mathcal{E}.\text{basis-indep indep-inD}$  by blast
  next
  fix x
  assume  $x \in \text{carrier} - B$ 
  then have  $\neg \text{indep-in carrier} (\text{insert } x B)$  using RTL  $\mathcal{E}.\text{basisD}$  by auto
  then show  $\neg \text{indep} (\text{insert } x B)$  using indep-subset-carrier indep-inI by blast

```

```

    qed
  qed
qed

context
  fixes  $\mathcal{E}$ 
  assumes  $\mathcal{E} \subseteq \text{carrier}$ 
begin

interpretation  $\mathcal{E}$ : indep-system  $\mathcal{E}$  indep-in  $\mathcal{E}$ 
  using  $\langle \mathcal{E} \subseteq \text{carrier} \rangle$  by auto

lemma basis-inI-aux:  $\mathcal{E}.\text{basis } X \implies \text{basis-in } \mathcal{E} X$ 
  unfolding basis-in-def by auto

lemma basis-inD-aux:  $\text{basis-in } \mathcal{E} X \implies \mathcal{E}.\text{basis } X$ 
  unfolding basis-in-def by auto

lemma not-basis-inD-aux:  $\neg \text{basis-in } \mathcal{E} X \implies \neg \mathcal{E}.\text{basis } X$ 
  using basis-inI-aux by auto

lemmas basis-inI = basis-inI-aux[OF  $\mathcal{E}.\text{basisI}$ ]
lemmas basis-in-indep-in =  $\mathcal{E}.\text{basis-indep}$ [OF basis-inD-aux]
lemmas basis-in-max-indep-in =  $\mathcal{E}.\text{basis-max-indep}$ [OF basis-inD-aux]
lemmas basis-inD =  $\mathcal{E}.\text{basisD}$ [OF basis-inD-aux]
lemmas basis-in-subset-carrier =  $\mathcal{E}.\text{basis-subset-carrier}$ [OF basis-inD-aux]
lemmas basis-in-finite =  $\mathcal{E}.\text{basis-finite}$ [OF basis-inD-aux]
lemmas indep-in-not-basis-in =  $\mathcal{E}.\text{indep-not-basis}$ [OF - not-basis-inD-aux]
lemmas basis-in-subset-eq =  $\mathcal{E}.\text{basis-subset-eq}$ [OF basis-inD-aux basis-inD-aux]

end

context
  fixes  $\mathcal{E}$ 
  assumes *:  $\mathcal{E} \subseteq \text{carrier}$ 
begin

interpretation  $\mathcal{E}$ : indep-system  $\mathcal{E}$  indep-in  $\mathcal{E}$ 
  using * by auto

lemma basis-in-sub-cong:
  assumes  $\mathcal{E}' \subseteq \mathcal{E}$ 
  shows  $\mathcal{E}.\text{basis-in } \mathcal{E}' B \longleftrightarrow \text{basis-in } \mathcal{E}' B$ 
proof (safe, goal-cases LTR RTL)
  case LTR
  show ?case
  proof (rule basis-inI)
    show  $\mathcal{E}' \subseteq \text{carrier}$  using assms * by auto
  next

```

```

    show indep-in  $\mathcal{E}' B$ 
    using * assms LTR  $\mathcal{E}$ .basis-in-subset-carrier  $\mathcal{E}$ .basis-in-indep-in indep-in-sub-cong
by auto
next
  fix  $x$ 
  assume  $x \in \mathcal{E}' - B$ 
  then show  $\neg indep-in \mathcal{E}' (insert\ x\ B)$ 
    using * assms LTR  $\mathcal{E}$ .basis-in-max-indep-in  $\mathcal{E}$ .basis-in-subset-carrier indep-in-sub-cong
by auto
  qed
next
  case RTL
  show ?case
  proof (rule  $\mathcal{E}$ .basis-inI)
    show  $\mathcal{E}' \subseteq \mathcal{E}$  using assms by auto
  next
    show  $\mathcal{E}$ .indep-in  $\mathcal{E}' B$ 
    using * assms RTL basis-in-subset-carrier basis-in-indep-in indep-in-sub-cong
by auto
next
  fix  $x$ 
  assume  $x \in \mathcal{E}' - B$ 
  then show  $\neg \mathcal{E}$ .indep-in  $\mathcal{E}' (insert\ x\ B)$ 
    using * assms RTL basis-in-max-indep-in basis-in-subset-carrier indep-in-sub-cong
by auto
  qed
qed
end

```

### 1.3 Circuits

A *circuit* is a minimal dependent set, i. e. a set which becomes independent on removing any element of the ground set.

**definition** *circuit* where  $circuit\ X \iff X \subseteq carrier \wedge \neg indep\ X \wedge (\forall x \in X. indep\ (X - \{x\}))$

**lemma** *circuitI*:

```

  assumes  $X \subseteq carrier$ 
  assumes  $\neg indep\ X$ 
  assumes  $\bigwedge x. x \in X \implies indep\ (X - \{x\})$ 
  shows circuit  $X$ 
  using assms unfolding circuit-def by auto

```

**lemma** *circuit-subset-carrier*:  $circuit\ X \implies X \subseteq carrier$

```

  unfolding circuit-def by auto

```

**lemmas** *circuit-finite* [*simp*] = *finite-subset*[*OF circuit-subset-carrier carrier-finite*]

**lemma** *circuit-dep*:  $circuit\ X \implies \neg indep\ X$

**unfolding** *circuit-def* **by** *auto*

**lemma** *circuit-min-dep*:  $\text{circuit } X \implies x \in X \implies \text{indep } (X - \{x\})$   
**unfolding** *circuit-def* **by** *auto*

**lemmas** *circuitD = circuit-subset-carrier circuit-dep circuit-min-dep*

**lemma** *circuit-nonempty*:  $\text{circuit } X \implies X \neq \{\}$   
**using** *circuit-dep indep-empty* **by** *blast*

**lemma** *dep-not-circuit*:  
**assumes**  $X \subseteq \text{carrier}$   
**assumes**  $\neg \text{indep } X$   
**assumes**  $\neg \text{circuit } X$   
**shows**  $\exists x \in X. \neg \text{indep } (X - \{x\})$   
**using** *assms circuitI* **by** *auto*

**lemma** *circuit-subset-eq*:  
**assumes** *circuit*  $C_1$   
**assumes** *circuit*  $C_2$   
**assumes**  $C_1 \subseteq C_2$   
**shows**  $C_1 = C_2$   
**proof** (*rule ccontr*)  
**assume**  $C_1 \neq C_2$   
**then obtain**  $x$  **where**  $x \notin C_1$   $x \in C_2$  **using** *assms* **by** *auto*  
**then have** *indep*  $C_1$  **using** *indep-subset*  $\langle C_1 \subseteq C_2 \rangle$  *circuit-min-dep* [*OF*  $\langle \text{circuit } C_2 \rangle$ , *of x*] **by** *auto*  
**then show** *False* **using** *assms circuitD* **by** *auto*  
**qed**

**definition** *circuit-in* **where**  
*circuit-in*  $\mathcal{E} X \longleftrightarrow \text{indep-system.circuit } \mathcal{E} (\text{indep-in } \mathcal{E}) X$

**context**  
**fixes**  $\mathcal{E}$   
**assumes**  $\mathcal{E} \subseteq \text{carrier}$   
**begin**

**interpretation**  $\mathcal{E}$ : *indep-system*  $\mathcal{E}$  *indep-in*  $\mathcal{E}$   
**using**  $\langle \mathcal{E} \subseteq \text{carrier} \rangle$  **by** *auto*

**lemma** *circuit-inI-aux*:  $\mathcal{E}.\text{circuit } X \implies \text{circuit-in } \mathcal{E} X$   
**unfolding** *circuit-in-def* **by** *auto*

**lemma** *circuit-inD-aux*:  $\text{circuit-in } \mathcal{E} X \implies \mathcal{E}.\text{circuit } X$   
**unfolding** *circuit-in-def* **by** *auto*

**lemma** *not-circuit-inD-aux*:  $\neg \text{circuit-in } \mathcal{E} X \implies \neg \mathcal{E}.\text{circuit } X$   
**using** *circuit-inI-aux* **by** *auto*

```

lemmas circuit-inI = circuit-inI-ax[OF  $\mathcal{E}$ .circuitI]

lemmas circuit-in-subset-carrier =  $\mathcal{E}$ .circuit-subset-carrier[OF circuit-inD-ax]
lemmas circuit-in-finite =  $\mathcal{E}$ .circuit-finite[OF circuit-inD-ax]
lemmas circuit-in-dep-in =  $\mathcal{E}$ .circuit-dep[OF circuit-inD-ax]
lemmas circuit-in-min-dep-in =  $\mathcal{E}$ .circuit-min-dep[OF circuit-inD-ax]
lemmas circuit-inD =  $\mathcal{E}$ .circuitD[OF circuit-inD-ax]
lemmas circuit-in-nonempty =  $\mathcal{E}$ .circuit-nonempty[OF circuit-inD-ax]
lemmas dep-in-not-circuit-in =  $\mathcal{E}$ .dep-not-circuit[OF - - not-circuit-inD-ax]
lemmas circuit-in-subset-eq =  $\mathcal{E}$ .circuit-subset-eq[OF circuit-inD-ax circuit-inD-ax]

```

**end**

```

lemma circuit-in-subI:
  assumes  $\mathcal{E}' \subseteq \mathcal{E}$   $\mathcal{E} \subseteq \text{carrier}$ 
  assumes circuit-in  $\mathcal{E}'$   $C$ 
  shows circuit-in  $\mathcal{E}$   $C$ 
proof (rule circuit-inI)
  show  $\mathcal{E} \subseteq \text{carrier}$  using assms by auto
next
  show  $C \subseteq \mathcal{E}$  using assms circuit-in-subset-carrier[of  $\mathcal{E}'$   $C$ ] by auto
next
  show  $\neg \text{indep-in}$   $\mathcal{E}$   $C$ 
  using assms
  circuit-in-dep-in[where  $\mathcal{E} = \mathcal{E}'$  and  $X = C$ ]
  circuit-in-subset-carrier dep-in-subI[where  $\mathcal{E}' = \mathcal{E}'$  and  $\mathcal{E} = \mathcal{E}$ ]
  by auto
next
fix  $x$ 
assume  $x \in C$ 
then show  $\text{indep-in}$   $\mathcal{E}$  ( $C - \{x\}$ )
  using assms circuit-in-min-dep-in indep-in-subI-subset by auto
qed

```

```

lemma circuit-in-supI:
  assumes  $\mathcal{E}' \subseteq \mathcal{E}$   $\mathcal{E} \subseteq \text{carrier}$   $C \subseteq \mathcal{E}'$ 
  assumes circuit-in  $\mathcal{E}$   $C$ 
  shows circuit-in  $\mathcal{E}'$   $C$ 
proof (rule circuit-inI)
  show  $\mathcal{E}' \subseteq \text{carrier}$  using assms by auto
next
  show  $C \subseteq \mathcal{E}'$  using assms by auto
next
  have  $\neg \text{indep-in}$   $\mathcal{E}$   $C$  using assms circuit-in-dep-in by auto
  then show  $\neg \text{indep-in}$   $\mathcal{E}'$   $C$  using assms dep-in-subI[of  $C$   $\mathcal{E}$ ] by auto
next
fix  $x$ 
assume  $x \in C$ 

```

**then have** *indep-in*  $\mathcal{E} (C - \{x\})$  **using** *assms circuit-in-min-dep-in* **by auto**  
**then have** *indep-in*  $\mathcal{E}' ((C - \{x\}) \cap \mathcal{E}')$  **using** *indep-in-subI* **by auto**  
**moreover have**  $(C - \{x\}) \cap \mathcal{E}' = C - \{x\}$  **using** *assms circuit-in-subset-carrier*  
**by auto**  
**ultimately show** *indep-in*  $\mathcal{E}' (C - \{x\})$  **by auto**  
**qed**

**context**  
**fixes**  $\mathcal{E}$   
**assumes** \*:  $\mathcal{E} \subseteq \text{carrier}$   
**begin**

**interpretation**  $\mathcal{E}$ : *indep-system*  $\mathcal{E}$  *indep-in*  $\mathcal{E}$   
**using** \* **by auto**

**lemma** *circuit-in-sub-cong*:  
**assumes**  $\mathcal{E}' \subseteq \mathcal{E}$   
**shows**  $\mathcal{E}.\text{circuit-in } \mathcal{E}' C \longleftrightarrow \text{circuit-in } \mathcal{E}' C$   
**proof** (*safe, goal-cases LTR RTL*)  
**case** *LTR*  
**show** ?*case*  
**proof** (*rule circuit-inI*)  
**show**  $\mathcal{E}' \subseteq \text{carrier}$  **using** *assms \** **by auto**  
**next**  
**show**  $C \subseteq \mathcal{E}'$   
**using** *assms LTR*  $\mathcal{E}.\text{circuit-in-subset-carrier}$  **by auto**  
**next**  
**show**  $\neg \text{indep-in } \mathcal{E}' C$   
**using** *assms LTR*  $\mathcal{E}.\text{circuit-in-dep-in indep-in-sub-cong}[OF *]$  **by auto**  
**next**  
**fix**  $x$   
**assume**  $x \in C$   
**then show** *indep-in*  $\mathcal{E}' (C - \{x\})$   
**using** *assms LTR*  $\mathcal{E}.\text{circuit-in-min-dep-in indep-in-sub-cong}[OF *]$  **by auto**  
**qed**  
**next**  
**case** *RTL*  
**show** ?*case*  
**proof** (*rule*  $\mathcal{E}.\text{circuit-inI}$ )  
**show**  $\mathcal{E}' \subseteq \mathcal{E}$  **using** *assms \** **by auto**  
**next**  
**show**  $C \subseteq \mathcal{E}'$   
**using** *assms \** *RTL circuit-in-subset-carrier* **by auto**  
**next**  
**show**  $\neg \mathcal{E}.\text{indep-in } \mathcal{E}' C$   
**using** *assms \** *RTL circuit-in-dep-in indep-in-sub-cong}[OF \*] **by auto**  
**next**  
**fix**  $x$   
**assume**  $x \in C$*

```

    then show  $\mathcal{E}.indep\text{-}in\ \mathcal{E}'\ (C - \{x\})$ 
      using assms * RTL circuit-in-min-dep-in indep-in-sub-cong[OF *] by auto
    qed
  qed
end

lemma circuit-imp-circuit-in:
  assumes circuit C
  shows circuit-in carrier C
proof (rule circuit-inI)
  show  $C \subseteq carrier$  using circuit-subset-carrier[OF assms] .
next
  show  $\neg indep\text{-}in\ carrier\ C$  using circuit-dep[OF assms] indep-in-indep by auto
next
  fix x
  assume  $x \in C$ 
  then have indep  $(C - \{x\})$  using circuit-min-dep[OF assms] by auto
  then show indep-in carrier  $(C - \{x\})$  using circuit-subset-carrier[OF assms]
by (auto intro: indep-inI)
qed auto

```

#### 1.4 Relation between independence and bases

A set is independent iff it is a subset of a basis.

```

lemma indep-imp-subset-basis:
  assumes indep X
  shows  $\exists B. basis\ B \wedge X \subseteq B$ 
  using assms
proof (induction X rule: psubset-inc-induct)
  case carrier
  show ?case using indep-subset-carrier[OF assms] .
next
  case (step X)
  {
    assume  $\neg basis\ X$ 
    then obtain x where  $x \in carrier\ x \notin X\ indep$  (insert x X)
      using step.premis indep-not-basis by auto
    then have ?case using step.IH[of insert x X] indep-subset-carrier by auto
  }
  then show ?case by auto
qed

```

lemmas *subset-basis-imp-indep = indep-subset*[*OF basis-indep*]

```

lemma indep-iff-subset-basis:  $indep\ X \longleftrightarrow (\exists B. basis\ B \wedge X \subseteq B)$ 
  using indep-imp-subset-basis subset-basis-imp-indep by auto

```

lemma *basis-ex*:  $\exists B. basis\ B$

```

using indep-imp-subset-basis[OF indep-empty] by auto

context
  fixes  $\mathcal{E}$ 
  assumes *:  $\mathcal{E} \subseteq \text{carrier}$ 
begin

interpretation  $\mathcal{E}$ : indep-system  $\mathcal{E}$  indep-in  $\mathcal{E}$ 
  using * by auto

lemma indep-in-imp-subset-basis-in:
  assumes indep-in  $\mathcal{E}$   $X$ 
  shows  $\exists B. \text{basis-in } \mathcal{E} B \wedge X \subseteq B$ 
  unfolding basis-in-def using  $\mathcal{E}.\text{indep-imp-subset-basis}$ [OF assms] .

lemmas subset-basis-in-imp-indep-in = indep-in-subset[OF * basis-in-indep-in[OF *]]

lemma indep-in-iff-subset-basis-in: indep-in  $\mathcal{E}$   $X \longleftrightarrow (\exists B. \text{basis-in } \mathcal{E} B \wedge X \subseteq B)$ 
  using indep-in-imp-subset-basis-in subset-basis-in-imp-indep-in by auto

lemma basis-in-ex:  $\exists B. \text{basis-in } \mathcal{E} B$ 
  unfolding basis-in-def using  $\mathcal{E}.\text{basis-ex}$  .

lemma basis-in-subI:
  assumes  $\mathcal{E}' \subseteq \mathcal{E}$   $\mathcal{E} \subseteq \text{carrier}$ 
  assumes basis-in  $\mathcal{E}' B$ 
  shows  $\exists B' \subseteq \mathcal{E} - \mathcal{E}'. \text{basis-in } \mathcal{E} (B \cup B')$ 
proof –
  have indep-in  $\mathcal{E} B$  using assms basis-in-indep-in indep-in-subI-subset by auto
  then obtain  $B'$  where  $B'$ : basis-in  $\mathcal{E} B' B \subseteq B'$ 
  using assms indep-in-imp-subset-basis-in[of  $B$ ] by auto
  show ?thesis
  proof (rule exI)
  have  $B' - B \subseteq \mathcal{E} - \mathcal{E}'$ 
  proof
  fix  $x$ 
  assume *:  $x \in B' - B$ 
  then have  $x \in \mathcal{E} x \notin B$ 
  using assms  $\langle \text{basis-in } \mathcal{E} B' \rangle$  basis-in-subset-carrier[of  $\mathcal{E}$ ] by auto
  moreover {
  assume  $x \in \mathcal{E}'$ 
  moreover have indep-in  $\mathcal{E}$  (insert  $x B$ )
  using * assms indep-in-subset[OF - basis-in-indep-in]  $B'$  by auto
  ultimately have indep-in  $\mathcal{E}'$  (insert  $x B$ )
  using assms basis-in-subset-carrier unfolding indep-in-def by auto
  then have False using assms *  $\langle x \in \mathcal{E}' \rangle$  basis-in-max-indep-in by auto
  }
  }

```

ultimately show  $x \in \mathcal{E} - \mathcal{E}'$  by auto  
 qed  
 moreover have  $B \cup (B' - B) = B'$  using  $\langle B \subseteq B' \rangle$  by auto  
 ultimately show  $B' - B \subseteq \mathcal{E} - \mathcal{E}' \wedge \text{basis-in } \mathcal{E} (B \cup (B' - B))$   
 using  $\langle \text{basis-in } \mathcal{E} B' \rangle$  by auto  
 qed  
 qed  
  
 lemma *basis-in-supI*:  
 assumes  $B \subseteq \mathcal{E}' \ \mathcal{E}' \subseteq \mathcal{E} \ \mathcal{E} \subseteq \text{carrier}$   
 assumes *basis-in*  $\mathcal{E} B$   
 shows *basis-in*  $\mathcal{E}' B$   
 proof (rule *basis-inI*)  
 show  $\mathcal{E}' \subseteq \text{carrier}$  using *assms* by auto  
 next  
 show *indep-in*  $\mathcal{E}' B$   
 proof -  
 have *indep-in*  $\mathcal{E}' (B \cap \mathcal{E}')$   
 using *assms* *basis-in-indep-in*[of  $\mathcal{E} B$ ] *indep-in-subI* by auto  
 moreover have  $B \cap \mathcal{E}' = B$  using *assms* by auto  
 ultimately show *?thesis* by auto  
 qed  
 next  
 show  $\bigwedge x. x \in \mathcal{E}' - B \implies \neg \text{indep-in } \mathcal{E}' (\text{insert } x B)$   
 using *assms* *basis-in-subset-carrier* *basis-in-max-indep-in* *dep-in-subI*[of  $\mathcal{E} \ \mathcal{E}'$ ]  
 by auto  
 qed  
  
 end

## 1.5 Relation between dependence and circuits

A set is dependent iff it contains a circuit.

lemma *dep-imp-supset-circuit*:  
 assumes  $X \subseteq \text{carrier}$   
 assumes  $\neg \text{indep } X$   
 shows  $\exists C. \text{circuit } C \wedge C \subseteq X$   
 using *assms*  
 proof (induction  $X$  rule: *remove-induct*)  
 case (remove  $X$ )  
 {  
 assume  $\neg \text{circuit } X$   
 then obtain  $x$  where  $x \in X \wedge \neg \text{indep } (X - \{x\})$   
 using *remove.prem*s *dep-not-circuit* by auto  
 then obtain  $C$  where *circuit*  $C \ C \subseteq X - \{x\}$   
 using *remove.prem*s *remove.IH*[of  $x$ ] by auto  
 then have *?case* by auto  
 }  
 then show *?case* using *remove.prem*s by auto

**qed** (*auto simp add: carrier-finite finite-subset*)

**lemma** *supset-circuit-imp-dep*:  
 **assumes** *circuit C ∧ C ⊆ X*  
 **shows**  $\neg$  *indep X*  
 **using** *assms indep-subset circuit-dep* **by** *auto*

**lemma** *dep-iff-supset-circuit*:  
 **assumes**  $X \subseteq$  *carrier*  
 **shows**  $\neg$  *indep X*  $\longleftrightarrow$   $(\exists C. \text{circuit } C \wedge C \subseteq X)$   
 **using** *assms dep-imp-supset-circuit supset-circuit-imp-dep* **by** *auto*

**context**  
 **fixes**  $\mathcal{E}$   
 **assumes**  $\mathcal{E} \subseteq$  *carrier*  
**begin**

**interpretation**  $\mathcal{E}$ : *indep-system*  $\mathcal{E}$  *indep-in*  $\mathcal{E}$   
 **using**  $\langle \mathcal{E} \subseteq$  *carrier*  $\rangle$  **by** *auto*

**lemma** *dep-in-imp-supset-circuit-in*:  
 **assumes**  $X \subseteq \mathcal{E}$   
 **assumes**  $\neg$  *indep-in*  $\mathcal{E} X$   
 **shows**  $\exists C. \text{circuit-in } \mathcal{E} C \wedge C \subseteq X$   
 **unfolding** *circuit-in-def* **using**  $\mathcal{E}.\text{dep-imp-supset-circuit}$ [*OF assms*] .

**lemma** *supset-circuit-in-imp-dep-in*:  
 **assumes** *circuit-in*  $\mathcal{E} C \wedge C \subseteq X$   
 **shows**  $\neg$  *indep-in*  $\mathcal{E} X$   
 **using** *assms*  $\mathcal{E}.\text{supset-circuit-imp-dep}$  **unfolding** *circuit-in-def* **by** *auto*

**lemma** *dep-in-iff-supset-circuit-in*:  
 **assumes**  $X \subseteq \mathcal{E}$   
 **shows**  $\neg$  *indep-in*  $\mathcal{E} X$   $\longleftrightarrow$   $(\exists C. \text{circuit-in } \mathcal{E} C \wedge C \subseteq X)$   
 **using** *assms dep-in-imp-supset-circuit-in supset-circuit-in-imp-dep-in* **by** *auto*

**end**

## 1.6 Ranks

**definition** *lower-rank-of* :: 'a set  $\Rightarrow$  nat **where**  
 *lower-rank-of carrier'*  $\equiv$  *Min* {*card B* | *B. basis-in carrier' B*}

**definition** *upper-rank-of* :: 'a set  $\Rightarrow$  nat **where**  
 *upper-rank-of carrier'*  $\equiv$  *Max* {*card B* | *B. basis-in carrier' B*}

**lemma** *collect-basis-finite*: *finite* (*Collect basis*)  
**proof** –  
 **have** *Collect basis*  $\subseteq$  {*X. X ⊆ carrier*}

using *basis-subset-carrier* by *auto*  
 moreover have *finite* ...  
 using *carrier-finite* by *auto*  
 ultimately show *?thesis* using *finite-subset* by *auto*  
 qed

**context**  
 fixes  $\mathcal{E}$   
 assumes  $*$ :  $\mathcal{E} \subseteq \text{carrier}$   
**begin**

**interpretation**  $\mathcal{E}$ : *indep-system*  $\mathcal{E}$  *indep-in*  $\mathcal{E}$   
 using  $*$  by *auto*

**lemma** *collect-basis-in-finite*: *finite* (*Collect* (*basis-in*  $\mathcal{E}$ ))  
 unfolding *basis-in-def* using  $\mathcal{E}.\text{collect-basis-finite}$  .

**lemma** *lower-rank-of-le*: *lower-rank-of*  $\mathcal{E} \leq \text{card } \mathcal{E}$   
**proof** –  
 have  $\exists n \in \{\text{card } B \mid B. \text{basis-in } \mathcal{E} B\}. n \leq \text{card } \mathcal{E}$   
 using *card-mono*[*OF*  $\mathcal{E}.\text{carrier-finite}$  *basis-in-subset-carrier*[*OF*  $*$ ]] *basis-in-ex*[*OF*  $*$ ]  
 $*$ ] by *auto*  
 moreover have *finite*  $\{\text{card } B \mid B. \text{basis-in } \mathcal{E} B\}$   
 using *collect-basis-in-finite* by *auto*  
 ultimately show *?thesis*  
 unfolding *lower-rank-of-def* using *basis-ex* *Min-le-iff* by *auto*  
 qed

**lemma** *upper-rank-of-le*: *upper-rank-of*  $\mathcal{E} \leq \text{card } \mathcal{E}$   
**proof** –  
 have  $\forall n \in \{\text{card } B \mid B. \text{basis-in } \mathcal{E} B\}. n \leq \text{card } \mathcal{E}$   
 using *card-mono*[*OF*  $\mathcal{E}.\text{carrier-finite}$  *basis-in-subset-carrier*[*OF*  $*$ ]] by *auto*  
 then show *?thesis*  
 unfolding *upper-rank-of-def* using *basis-in-ex*[*OF*  $*$ ] *collect-basis-in-finite* by  
*auto*  
 qed

**context**  
 fixes  $\mathcal{E}'$   
 assumes  $**$ :  $\mathcal{E}' \subseteq \mathcal{E}$   
**begin**

**interpretation**  $\mathcal{E}'_1$ : *indep-system*  $\mathcal{E}'$  *indep-in*  $\mathcal{E}'$   
 using  $**$  by *auto*  
**interpretation**  $\mathcal{E}'_2$ : *indep-system*  $\mathcal{E}'$   $\mathcal{E}.\text{indep-in}$   $\mathcal{E}'$   
 using  $**$  by *auto*

**lemma** *lower-rank-of-sub-cong*:  
 shows  $\mathcal{E}.\text{lower-rank-of } \mathcal{E}' = \text{lower-rank-of } \mathcal{E}'$

```

proof –
  have  $\wedge B. \mathcal{E}'_1.basis\ B \longleftrightarrow \mathcal{E}'_2.basis\ B$ 
    using ** basis-in-sub-cong[OF *, of  $\mathcal{E}$ ]
    unfolding basis-in-def  $\mathcal{E}.basis-in-def$  by auto
  then show ?thesis
    unfolding lower-rank-of-def  $\mathcal{E}.lower-rank-of-def$ 
    using basis-in-sub-cong[OF * **]
    by auto
qed

```

```

lemma upper-rank-of-sub-cong:
  shows  $\mathcal{E}.upper-rank-of\ \mathcal{E}' = upper-rank-of\ \mathcal{E}'$ 
proof –
  have  $\wedge B. \mathcal{E}'_1.basis\ B \longleftrightarrow \mathcal{E}'_2.basis\ B$ 
    using ** basis-in-sub-cong[OF *, of  $\mathcal{E}$ ]
    unfolding basis-in-def  $\mathcal{E}.basis-in-def$  by auto
  then show ?thesis
    unfolding upper-rank-of-def  $\mathcal{E}.upper-rank-of-def$ 
    using basis-in-sub-cong[OF * **]
    by auto
qed

```

end

end

end

end

## 2 Matroids

```

theory Matroid
  imports Indep-System
begin

```

```

lemma card-subset-ex:
  assumes finite  $A$   $n \leq card\ A$ 
  shows  $\exists B \subseteq A. card\ B = n$ 
using assms
proof (induction  $A$  arbitrary;  $n$  rule: finite-induct)
  case (insert  $x\ A$ )
  show ?case
  proof (cases  $n$ )
    case 0
    then show ?thesis using card.empty by blast
  next
  case (Suc  $k$ )
  then have  $\exists B \subseteq A. card\ B = k$  using insert by auto

```

```

then obtain  $B$  where  $B \subseteq A$   $\text{card } B = k$  by auto
moreover from this have finite  $B$  using insert.hyps finite-subset by auto
ultimately have  $\text{card } (\text{insert } x B) = n$ 
  using Suc insert.hyps card-insert-disjoint by fastforce
then show ?thesis using  $\langle B \subseteq A \rangle$  by blast
qed
qed auto

locale matroid = indep-system +
  assumes augment-aux:
    indep  $X \implies$  indep  $Y \implies$   $\text{card } X = \text{Suc } (\text{card } Y) \implies \exists x \in X - Y. \text{ indep } (\text{insert } x Y)$ 
begin

lemma augment:
  assumes indep  $X$  indep  $Y$   $\text{card } Y < \text{card } X$ 
  shows  $\exists x \in X - Y. \text{ indep } (\text{insert } x Y)$ 
proof -
  obtain  $X'$  where  $X' \subseteq X$   $\text{card } X' = \text{Suc } (\text{card } Y)$ 
  using assms card-subset-ex[of  $X$   $\text{Suc } (\text{card } Y)$ ] indep-finite by auto
  then obtain  $x$  where  $x \in X' - Y$  indep  $(\text{insert } x Y)$ 
  using assms augment-aux[of  $X' Y$ ] indep-subset by auto
  then show ?thesis using  $\langle X' \subseteq X \rangle$  by auto
qed

lemma augment-psubset:
  assumes indep  $X$  indep  $Y$   $Y \subset X$ 
  shows  $\exists x \in X - Y. \text{ indep } (\text{insert } x Y)$ 
  using assms augment-psubset-card-mono indep-finite by blast

2.1 Minors

A subset of the ground set induces a matroid.

lemma matroid-subset [simp, intro]:
  assumes  $\mathcal{E} \subseteq \text{carrier}$ 
  shows matroid  $\mathcal{E}$  (indep-in  $\mathcal{E}$ )
  unfolding matroid-def matroid-axioms-def
proof (safe, goal-cases indep-system augment)
  case indep-system
  then show ?case using indep-system-subset[OF assms] .
next
  case (augment  $X Y$ )
  then show ?case using augment-aux[of  $X Y$ ] unfolding indep-in-def by auto
qed

context
  fixes  $\mathcal{E}$ 
  assumes  $\mathcal{E} \subseteq \text{carrier}$ 
begin

```

**interpretation**  $\mathcal{E}$ : *matroid*  $\mathcal{E}$  *indep-in*  $\mathcal{E}$   
**using**  $\langle \mathcal{E} \subseteq \text{carrier} \rangle$  **by** *auto*

**lemmas** *augment-aux-indep-in* =  $\mathcal{E}.\text{augment-aux}$   
**lemmas** *augment-indep-in* =  $\mathcal{E}.\text{augment}$   
**lemmas** *augment-psubset-indep-in* =  $\mathcal{E}.\text{augment-psubset}$

**end**

## 2.2 Bases

**lemma** *basis-card*:  
**assumes** *basis*  $B_1$   
**assumes** *basis*  $B_2$   
**shows**  $\text{card } B_1 = \text{card } B_2$   
**proof** (*rule ccontr, goal-cases False*)  
**case** *False*  
**then have**  $\text{card } B_1 < \text{card } B_2 \vee \text{card } B_2 < \text{card } B_1$  **by** *auto*  
**moreover** {  
**fix**  $B_1 B_2$   
**assume** *basis*  $B_1$  *basis*  $B_2$   $\text{card } B_1 < \text{card } B_2$   
**then obtain**  $x$  **where**  $x \in B_2 - B_1$  *indep* (*insert*  $x B_1$ )  
**using** *augment basisD* **by** *blast*  
**then have**  $x \in \text{carrier} - B_1$   
**using**  $\langle \text{basis } B_1 \rangle$  *basisD indep-subset-carrier* **by** *blast*  
**then have**  $\neg \text{indep} (\text{insert } x B_1)$  **using**  $\langle \text{basis } B_1 \rangle$  *basisD* **by** *auto*  
**then have** *False* **using**  $\langle \text{indep} (\text{insert } x B_1) \rangle$  **by** *auto*  
**}**  
**ultimately show** *?case* **using** *assms* **by** *auto*  
**qed**

**lemma** *basis-indep-card*:  
**assumes** *indep*  $X$   
**assumes** *basis*  $B$   
**shows**  $\text{card } X \leq \text{card } B$   
**proof** –  
**obtain**  $B'$  **where** *basis*  $B'$   $X \subseteq B'$  **using** *assms indep-imp-subset-basis* **by** *auto*  
**then show** *?thesis* **using** *assms basis-finite basis-card*[*of*  $B B'$ ] **by** (*auto intro: card-mono*)  
**qed**

**lemma** *basis-augment*:  
**assumes** *basis*  $B_1$  *basis*  $B_2$   $x \in B_1 - B_2$   
**shows**  $\exists y \in B_2 - B_1.$  *basis* (*insert*  $y (B_1 - \{x\})$ )  
**proof** –  
**let**  $?B_1 = B_1 - \{x\}$   
**have**  $\text{card } ?B_1 < \text{card } B_2$   
**using** *assms basis-card*[*of*  $B_1 B_2$ ] *card-Diff1-less*[*OF basis-finite, of*  $B_1$ ] **by** *auto*

**moreover have**  $\text{indep } ?B_1$  **using**  $\text{assms basis-indep[of } B_1] \text{ indep-subset[of } B_1 ?B_1]$  **by auto**  
**ultimately obtain**  $y$  **where**  $y: y \in B_2 - ?B_1$   $\text{indep (insert } y ?B_1)$   
**using**  $\text{assms augment[of } B_2 ?B_1] \text{ basis-indep}$  **by auto**  
**let**  $?B_1' = \text{insert } y ?B_1$   
**have**  $\text{basis } ?B_1'$  **using**  $\langle \text{indep } ?B_1' \rangle$   
**proof** ( $\text{rule basisI, goal-cases insert}$ )  
**case** ( $\text{insert } x$ )  
**have**  $\text{card (insert } x ?B_1') > \text{card } B_1$   
**proof** –  
**have**  $\text{card (insert } x ?B_1') = \text{Suc (card } ?B_1')$   
**using**  $\text{insert card.insert-remove[OF indep-finite, of } ?B_1'] y$  **by auto**  
**also have**  $\dots = \text{Suc (Suc (card } ?B_1))$   
**using**  $\text{card.insert-remove[OF indep-finite, of } ?B_1] \langle \text{indep } ?B_1' \rangle y$  **by auto**  
**also have**  $\dots = \text{Suc (card } B_1)$   
**using**  $\text{assms basis-finite[of } B_1] \text{ card.remove[of } B_1]$  **by auto**  
**finally show**  $?thesis$  **by auto**  
**qed**  
**then have**  $\neg \text{indep (insert } x (\text{insert } y ?B_1))$   
**using**  $\text{assms basis-indep-card[of insert } x (\text{insert } y ?B_1) B_1]$  **by auto**  
**moreover have**  $\text{insert } x (\text{insert } y ?B_1) \subseteq \text{carrier}$   
**using**  $\text{assms insert } y \text{ basis-finite indep-subset-carrier}$  **by auto**  
**ultimately show**  $?case$  **by auto**  
**qed**  
**then show**  $?thesis$  **using**  $\text{assms } y$  **by auto**  
**qed**

**context**  
**fixes**  $\mathcal{E}$   
**assumes**  $*: \mathcal{E} \subseteq \text{carrier}$   
**begin**

**interpretation**  $\mathcal{E}$ : *matroid*  $\mathcal{E}$  *indep-in*  $\mathcal{E}$   
**using**  $\langle \mathcal{E} \subseteq \text{carrier} \rangle$  **by auto**

**lemmas**  $\text{basis-in-card} = \mathcal{E}.\text{basis-card[OF basis-inD-aux[OF *] basis-inD-aux[OF *]]}$

**lemmas**  $\text{basis-in-indep-in-card} = \mathcal{E}.\text{basis-indep-card[OF - basis-inD-aux[OF *]]}$

**lemma** *basis-in-augment*:

**assumes**  $\text{basis-in } \mathcal{E} B_1 \text{ basis-in } \mathcal{E} B_2 x \in B_1 - B_2$   
**shows**  $\exists y \in B_2 - B_1. \text{basis-in } \mathcal{E} (\text{insert } y (B_1 - \{x\}))$   
**using**  $\text{assms } \mathcal{E}.\text{basis-augment unfolding basis-in-def}$  **by auto**

**end**

## 2.3 Circuits

**lemma** *circuit-elim*:

**assumes** *circuit*  $C_1$  *circuit*  $C_2$   $C_1 \neq C_2$   $x \in C_1 \cap C_2$   
**shows**  $\exists C_3 \subseteq (C_1 \cup C_2) - \{x\}$ . *circuit*  $C_3$   
**proof** –  
**let**  $?C = (C_1 \cup C_2) - \{x\}$   
**let**  $?carrier = C_1 \cup C_2$

**have** *assms'*: *circuit-in carrier*  $C_1$  *circuit-in carrier*  $C_2$   
**using** *assms circuit-imp-circuit-in* **by** *auto*

**have**  $?C \subseteq carrier$  **using** *assms circuit-subset-carrier* **by** *auto*  
**show** *?thesis*  
**proof** (*cases indep ?C*)  
**case** *False*  
**then show** *?thesis* **using** *dep-iff-supset-circuit*  $\langle ?C \subseteq carrier \rangle$  **by** *auto*  
**next**  
**case** *True*  
**then have** *indep-in ?carrier ?C* **using**  $\langle ?C \subseteq carrier \rangle$  **by** (*auto intro: indep-inI*)

**have**  $*$ :  $?carrier \subseteq carrier$  **using** *assms circuit-subset-carrier* **by** *auto*  
**obtain**  $y$  **where**  $y: y \in C_2$   $y \notin C_1$  **using** *assms circuit-subset-eq* **by** *blast*  
**then have** *indep-in ?carrier*  $(C_2 - \{y\})$   
**using** *assms' circuit-in-min-dep-in* $[OF * circuit-in-supI[OF *, of C_2]]$  **by**  
*auto*

**then obtain**  $B$  **where**  $B$ : *basis-in ?carrier*  $B$   $C_2 - \{y\} \subseteq B$   
**using**  $*$  *assms indep-in-imp-subset-basis-in* $[of ?carrier C_2 - \{y\}]$  **by** *auto*

**have**  $y \notin B$   
**proof** (*rule ccontr, goal-cases False*)  
**case** *False*  
**then have**  $C_2 \subseteq B$  **using**  $B$  **by** *auto*  
**moreover have** *circuit-in ?carrier*  $C_2$  **using**  $*$  *assms' circuit-in-supI* **by** *auto*  
**ultimately have**  $\neg indep-in ?carrier B$   
**using**  $B$  *basis-in-subset-carrier* $[OF *]$  *supset-circuit-in-imp-dep-in* $[OF *]$  **by**  
*auto*

**then show** *False* **using** *assms B basis-in-indep-in* $[OF *]$  **by** *auto*  
**qed**

**have**  $C_1 - B \neq \{\}$   
**proof** (*rule ccontr, goal-cases False*)  
**case** *False*  
**then have**  $C_1 - (C_1 \cap B) = \{\}$  **by** *auto*  
**then have**  $C_1 = C_1 \cap B$  **using** *assms circuit-subset-eq* **by** *auto*  
**moreover have** *indep*  $(C_1 \cap B)$   
**using** *assms B basis-in-indep-in* $[OF *]$  *indep-in-subset* $[OF *, of B C_1 \cap B]$   
*indep-in-indep*  
**by** *auto*  
**ultimately show** *?case* **using** *assms circuitD* **by** *auto*  
**qed**  
**then obtain**  $z$  **where**  $z: z \in C_1$   $z \notin B$  **by** *auto*

```

have  $y \neq z$  using  $y z$  by auto
have  $x \in C_1 \ x \in C_2$  using assms by auto

have finite ?carrier using assms carrier-finite finite-subset by auto
have  $\text{card } B \leq \text{card } (?carrier - \{y, z\})$ 
proof (rule card-mono)
  show finite ( $C_1 \cup C_2 - \{y, z\}$ ) using  $\langle \text{finite } ?carrier \rangle$  by auto
next
show  $B \subseteq C_1 \cup C_2 - \{y, z\}$ 
  using B basis-in-subset-carrier[OF *, of B]  $\langle y \notin B \rangle \langle z \notin B \rangle$  by auto
qed
also have  $\dots = \text{card } ?carrier - 2$ 
  using  $\langle \text{finite } ?carrier \rangle \langle y \in C_2 \rangle \langle z \in C_1 \rangle \langle y \neq z \rangle$  card-Diff-subset-Int by
auto
also have  $\dots < \text{card } ?carrier - 1$ 
proof -
  have  $\text{card } ?carrier = \text{card } C_1 + \text{card } C_2 - \text{card } (C_1 \cap C_2)$ 
    using assms  $\langle \text{finite } ?carrier \rangle$  card-Un-Int[of C1 C2] by auto
  also have  $\dots = \text{card } C_1 + (\text{card } C_2 - \text{card } (C_1 \cap C_2))$ 
    using assms  $\langle \text{finite } ?carrier \rangle$  card-mono[of C2] by auto
  also have  $\dots = \text{card } C_1 + \text{card } (C_2 - C_1)$ 
  proof -
    have  $\text{card } (C_2 - C_1) = \text{card } C_2 - \text{card } (C_2 \cap C_1)$ 
      using assms  $\langle \text{finite } ?carrier \rangle$  card-Diff-subset-Int[of C2 C1] by auto
    also have  $\dots = \text{card } C_2 - \text{card } (C_1 \cap C_2)$  by (simp add: inf-commute)
    finally show ?thesis by auto
  qed
  finally have  $\text{card } (C_1 \cup C_2) = \text{card } C_1 + \text{card } (C_2 - C_1)$  .
  moreover have  $\text{card } C_1 > 0$  using assms circuit-nonempty  $\langle \text{finite } ?carrier \rangle$ 
by auto
  moreover have  $\text{card } (C_2 - C_1) > 0$  using assms  $\langle \text{finite } ?carrier \rangle \langle y \in C_2 \rangle$ 
 $\langle y \notin C_1 \rangle$  by auto
  ultimately show ?thesis by auto
qed
also have  $\dots = \text{card } ?C$ 
  using  $\langle \text{finite } ?carrier \rangle$  card-Diff-singleton  $\langle x \in C_1 \rangle \langle x \in C_2 \rangle$  by auto
finally have  $\text{card } B < \text{card } ?C$  .
then have False
  using basis-in-indep-in-card[OF *, of ?C B] B  $\langle \text{indep-in } ?carrier ?C \rangle$  by auto
then show ?thesis by auto
qed
qed

lemma min-dep-imp-supset-circuit:
  assumes indep X
  assumes circuit C
  assumes  $C \subseteq \text{insert } x X$ 
  shows  $x \in C$ 

```

**proof** (*rule ccontr*)  
**assume**  $x \notin C$   
**then have**  $C \subseteq X$  **using** *assms* **by** *auto*  
**then have** *indep*  $C$  **using** *assms indep-subset* **by** *auto*  
**then show** *False* **using** *assms circuitD* **by** *auto*  
**qed**

**lemma** *min-dep-imp-ex1-supset-circuit*:

**assumes**  $x \in \text{carrier}$   
**assumes** *indep*  $X$   
**assumes**  $\neg \text{indep}$  (*insert*  $x$   $X$ )  
**shows**  $\exists! C. \text{circuit } C \wedge C \subseteq \text{insert } x$   $X$   
**proof** –  
**obtain**  $C$  **where**  $C: \text{circuit } C \wedge C \subseteq \text{insert } x$   $X$   
**using** *assms indep-subset-carrier dep-iff-supset-circuit* **by** *auto*

**show** *?thesis*

**proof** (*rule ex1I, goal-cases ex unique*)  
**show** *circuit*  $C \wedge C \subseteq \text{insert } x$   $X$  **using**  $C$  **by** *auto*  
**next**  
{  
**fix**  $C'$   
**assume**  $C': \text{circuit } C' \wedge C' \subseteq \text{insert } x$   $X$   
**have**  $C' = C$   
**proof** (*rule ccontr*)  
**assume**  $C' \neq C$   
**moreover have**  $x \in C' \cap C$  **using**  $C$   $C'$  *assms min-dep-imp-supset-circuit*  
**by** *auto*  
**ultimately have**  $\neg \text{indep}$  ( $C' \cup C - \{x\}$ )  
**using** *circuit-elim[OF C(1) C'(1), of x] supset-circuit-imp-dep[of - C' \cup C - \{x\}]* **by** *auto*  
**moreover have**  $C' \cup C - \{x\} \subseteq X$  **using**  $C$   $C'$  **by** *auto*  
**ultimately show** *False* **using** *assms indep-subset* **by** *auto*  
**qed**  
}  
**then show**  $\bigwedge C'. \text{circuit } C' \wedge C' \subseteq \text{insert } x$   $X \implies C' = C$   
**by** *auto*  
**qed**  
**qed**

**lemma** *basis-ex1-supset-circuit*:

**assumes** *basis*  $B$   
**assumes**  $x \in \text{carrier} - B$   
**shows**  $\exists! C. \text{circuit } C \wedge C \subseteq \text{insert } x$   $B$   
**using** *assms min-dep-imp-ex1-supset-circuit basisD* **by** *auto*

**definition** *fund-circuit* :: ' $a \Rightarrow 'a$  set  $\Rightarrow 'a$  set **where**

*fund-circuit*  $x$   $B \equiv (\text{THE } C. \text{circuit } C \wedge C \subseteq \text{insert } x$   $B)$

**lemma** *circuit-iff-fund-circuit*:  
*circuit*  $C \longleftrightarrow (\exists x B. x \in \text{carrier} - B \wedge \text{basis } B \wedge C = \text{fund-circuit } x B)$   
**proof** (*safe, goal-cases LTR RTL*)  
**case** *LTR*  
**then obtain**  $x$  **where**  $x \in C$  **using** *circuit-nonempty* **by** *auto*  
**then have** *indep*  $(C - \{x\})$  **using** *LTR unfolding circuit-def* **by** *auto*  
**then obtain**  $B$  **where**  $B: \text{basis } B \ C - \{x\} \subseteq B$  **using** *indep-imp-subset-basis*  
**by** *auto*  
**then have**  $x \in \text{carrier}$  **using** *LTR circuit-subset-carrier*  $\langle x \in C \rangle$  **by** *auto*  
**moreover have**  $x \notin B$   
**proof** (*rule ccontr, goal-cases False*)  
**case** *False*  
**then have**  $C \subseteq B$  **using**  $\langle C - \{x\} \subseteq B \rangle$  **by** *auto*  
**then have**  $\neg \text{indep } B$  **using** *LTR B basis-subset-carrier supset-circuit-imp-dep*  
**by** *auto*  
**then show** *?case* **using** *B basis-indep* **by** *auto*  
**qed**  
**ultimately show** *?case*  
**unfolding** *fund-circuit-def*  
**using** *LTR B theI-unique[OF basis-ex1-supset-circuit[of B x], of C]* **by** *auto*  
**next**  
**case** (*RTL x B*)  
**then have**  $\exists! C. \text{circuit } C \wedge C \subseteq \text{insert } x B$   
**using** *min-dep-imp-ex1-supset-circuit basisD[of B]* **by** *auto*  
**then show** *?case*  
**unfolding** *fund-circuit-def*  
**using** *theI[of  $\lambda C. \text{circuit } C \wedge C \subseteq \text{insert } x B$ ]* **by** *fastforce*  
**qed**

**lemma** *fund-circuitI*:  
**assumes** *basis B*  
**assumes**  $x \in \text{carrier} - B$   
**assumes** *circuit C*  
**assumes**  $C \subseteq \text{insert } x B$   
**shows** *fund-circuit*  $x B = C$   
**unfolding** *fund-circuit-def*  
**using** *assms theI-unique[OF basis-ex1-supset-circuit, of B x C]* **by** *auto*

**definition** *fund-circuit-in* **where** *fund-circuit-in*  $\mathcal{E} \ x B \equiv \text{matroid.fund-circuit } \mathcal{E}$   
*(indep-in  $\mathcal{E}$ ) x B*

**context**  
**fixes**  $\mathcal{E}$   
**assumes**  $*$ :  $\mathcal{E} \subseteq \text{carrier}$   
**begin**

**interpretation**  $\mathcal{E}$ : *matroid  $\mathcal{E}$  indep-in  $\mathcal{E}$*   
**using**  $\langle \mathcal{E} \subseteq \text{carrier} \rangle$  **by** *auto*

**lemma** *fund-circuit-inI-aux*:  $\mathcal{E}.fund-circuit\ x\ B = fund-circuit-in\ \mathcal{E}\ x\ B$   
**unfolding** *fund-circuit-in-def* **by** *auto*

**lemma** *circuit-in-elim*:  
**assumes** *circuit-in*  $\mathcal{E}\ C_1$  *circuit-in*  $\mathcal{E}\ C_2$   $C_1 \neq C_2$   $x \in C_1 \cap C_2$   
**shows**  $\exists C_3 \subseteq (C_1 \cup C_2) - \{x\}. circuit-in\ \mathcal{E}\ C_3$   
**using** *assms*  $\mathcal{E}.circuit-elim$  **unfolding** *circuit-in-def* **by** *auto*

**lemmas** *min-dep-in-imp-supset-circuit-in* =  $\mathcal{E}.min-dep-imp-supset-circuit[OF - circuit-inD-aux[OF *]]$

**lemma** *min-dep-in-imp-ex1-supset-circuit-in*:  
**assumes**  $x \in \mathcal{E}$   
**assumes** *indep-in*  $\mathcal{E}\ X$   
**assumes**  $\neg indep-in\ \mathcal{E}\ (insert\ x\ X)$   
**shows**  $\exists! C. circuit-in\ \mathcal{E}\ C \wedge C \subseteq insert\ x\ X$   
**using** *assms*  $\mathcal{E}.min-dep-imp-ex1-supset-circuit$  **unfolding** *circuit-in-def* **by** *auto*

**lemma** *basis-in-ex1-supset-circuit-in*:  
**assumes** *basis-in*  $\mathcal{E}\ B$   
**assumes**  $x \in \mathcal{E} - B$   
**shows**  $\exists! C. circuit-in\ \mathcal{E}\ C \wedge C \subseteq insert\ x\ B$   
**using** *assms*  $\mathcal{E}.basis-ex1-supset-circuit$  **unfolding** *circuit-in-def* *basis-in-def* **by** *auto*

**lemma** *fund-circuit-inI*:  
**assumes** *basis-in*  $\mathcal{E}\ B$   
**assumes**  $x \in \mathcal{E} - B$   
**assumes** *circuit-in*  $\mathcal{E}\ C$   
**assumes**  $C \subseteq insert\ x\ B$   
**shows** *fund-circuit-in*  $\mathcal{E}\ x\ B = C$   
**using** *assms*  $\mathcal{E}.fund-circuitI$   
**unfolding** *basis-in-def* *circuit-in-def* *fund-circuit-in-def* **by** *auto*

**end**

**context**  
**fixes**  $\mathcal{E}$   
**assumes**  $*$ :  $\mathcal{E} \subseteq carrier$   
**begin**

**interpretation**  $\mathcal{E}$ : *matroid*  $\mathcal{E}$  *indep-in*  $\mathcal{E}$   
**using**  $\langle \mathcal{E} \subseteq carrier \rangle$  **by** *auto*

**lemma** *fund-circuit-in-sub-cong*:  
**assumes**  $\mathcal{E}' \subseteq \mathcal{E}$   
**assumes**  $x \in \mathcal{E}' - B$   
**assumes** *basis-in*  $\mathcal{E}'\ B$   
**shows**  $\mathcal{E}.fund-circuit-in\ \mathcal{E}'\ x\ B = fund-circuit-in\ \mathcal{E}'\ x\ B$

**proof** –  
**obtain**  $C$  **where**  $C$ : *circuit-in*  $\mathcal{E}'$   $C$   $C \subseteq \text{insert } x B$   
**using** \* *assms* *basis-in-ex1-supset-circuit-in*[of  $\mathcal{E}' B x$ ] **by** *auto*  
**then have** *fund-circuit-in*  $\mathcal{E}' x B = C$   
**using** \* *assms* *fund-circuit-inI* **by** *auto*  
**also have**  $\dots = \mathcal{E}.\text{fund-circuit-in } \mathcal{E}' x B$   
**using** \* *assms*  $C \mathcal{E}.\text{fund-circuit-inI}$  *basis-in-sub-cong*[of  $\mathcal{E}$ ] *circuit-in-sub-cong*[of  $\mathcal{E}$ ] **by** *auto*  
**finally show** *?thesis* **by** *auto*  
**qed**  
**end**

## 2.4 Ranks

**abbreviation** *rank-of* **where**  $\text{rank-of} \equiv \text{lower-rank-of}$

**lemmas** *rank-of-def* = *lower-rank-of-def*

**lemmas** *rank-of-sub-cong* = *lower-rank-of-sub-cong*

**lemmas** *rank-of-le* = *lower-rank-of-le*

**context**

**fixes**  $\mathcal{E}$

**assumes** \*:  $\mathcal{E} \subseteq \text{carrier}$

**begin**

**interpretation**  $\mathcal{E}$ : *matroid*  $\mathcal{E}$  *indep-in*  $\mathcal{E}$

**using** \* **by** *auto*

**lemma** *lower-rank-of-eq-upper-rank-of*:  $\text{lower-rank-of } \mathcal{E} = \text{upper-rank-of } \mathcal{E}$

**proof** –

**obtain**  $B$  **where** *basis-in*  $\mathcal{E} B$  **using** *basis-in-ex*[ $OF$  \*] **by** *auto*

**then have**  $\{\text{card } B \mid B. \text{basis-in } \mathcal{E} B\} = \{\text{card } B\}$

**by** *safe* (*auto dest: basis-in-card*[ $OF$  \*])

**then show** *?thesis* **unfolding** *lower-rank-of-def* *upper-rank-of-def* **by** *auto*

**qed**

**lemma** *rank-of-eq-card-basis-in*:

**assumes** *basis-in*  $\mathcal{E} B$

**shows**  $\text{rank-of } \mathcal{E} = \text{card } B$

**proof** –

**have**  $\{\text{card } B \mid B. \text{basis-in } \mathcal{E} B\} = \{\text{card } B\}$  **using** *assms* **by** *safe* (*auto dest: basis-in-card*[ $OF$  \*])

**then show** *?thesis* **unfolding** *rank-of-def* **by** *auto*

**qed**

**lemma** *rank-of-indep-in-le*:

**assumes** *indep-in*  $\mathcal{E} X$

**shows**  $\text{card } X \leq \text{rank-of } \mathcal{E}$

**proof** –  
{  
  **fix**  $B$   
  **assume**  $\text{basis-in } \mathcal{E} B$   
  **moreover obtain**  $B'$  **where**  $\text{basis-in } \mathcal{E} B' X \subseteq B'$   
  **using**  $\text{assms indep-in-imp-subset-basis-in}[OF *]$  **by**  $\text{auto}$   
  **ultimately have**  $\text{card } X \leq \text{card } B$   
  **using**  $\text{card-mono}[OF \text{basis-in-finite}[OF *]] \text{basis-in-card}[OF *, \text{of } B B']$  **by**  
 $\text{auto}$   
}  
**moreover have**  $\text{finite } \{\text{card } B \mid B. \text{basis-in } \mathcal{E} B\}$   
**using**  $\text{collect-basis-in-finite}[OF *]$  **by**  $\text{auto}$   
**ultimately show**  $?thesis$   
**unfolding**  $\text{rank-of-def}$  **using**  $\text{basis-in-ex}[OF *]$  **by**  $\text{auto}$   
**qed**

**end**

**lemma**  $\text{rank-of-mono}$ :

**assumes**  $X \subseteq Y$   
**assumes**  $Y \subseteq \text{carrier}$   
**shows**  $\text{rank-of } X \leq \text{rank-of } Y$

**proof** –

**obtain**  $B_X$  **where**  $B_X: \text{basis-in } X B_X$  **using**  $\text{assms basis-in-ex}[of X]$  **by**  $\text{auto}$   
**moreover obtain**  $B_Y$  **where**  $B_Y: \text{basis-in } Y B_Y$  **using**  $\text{assms basis-in-ex}[of Y]$   
**by**  $\text{auto}$   
**moreover have**  $\text{card } B_X \leq \text{card } B_Y$   
**using**  $\text{assms basis-in-indep-in-card}[OF - - B_Y] \text{basis-in-indep-in}[OF - B_X]$   
 $\text{indep-in-subI-subset}$   
**by**  $\text{auto}$   
**ultimately show**  $?thesis$  **using**  $\text{assms rank-of-eq-card-basis-in}$  **by**  $\text{auto}$   
**qed**

**lemma**  $\text{rank-of-insert-le}$ :

**assumes**  $X \subseteq \text{carrier}$   
**assumes**  $x \in \text{carrier}$   
**shows**  $\text{rank-of } (\text{insert } x X) \leq \text{Suc } (\text{rank-of } X)$

**proof** –

**obtain**  $B$  **where**  $B: \text{basis-in } X B$  **using**  $\text{assms basis-in-ex}[of X]$  **by**  $\text{auto}$   
**have**  $\text{basis-in } (\text{insert } x X) B \vee \text{basis-in } (\text{insert } x X) (\text{insert } x B)$

**proof** –

**obtain**  $B'$  **where**  $B': B' \subseteq \text{insert } x X - X \text{basis-in } (\text{insert } x X) (B \cup B')$   
**using**  $\text{assms } B \text{basis-in-subI}[of \text{insert } x X X B]$  **by**  $\text{auto}$

**then have**  $B' = \{\} \vee B' = \{x\}$  **by**  $\text{auto}$

**then show**  $?thesis$

**proof**

**assume**  $B' = \{\}$   
**then have**  $\text{basis-in } (\text{insert } x X) B$  **using**  $B'$  **by**  $\text{auto}$   
**then show**  $?thesis$  **by**  $\text{auto}$

```

next
  assume  $B' = \{x\}$ 
  then have basis-in (insert  $x$   $X$ ) (insert  $x$   $B$ ) using  $B'$  by auto
  then show ?thesis by auto
qed
qed
then show ?thesis
proof
  assume basis-in (insert  $x$   $X$ )  $B$ 
  then show ?thesis
    using assms  $B$  rank-of-eq-card-basis-in by auto
next
  assume basis-in (insert  $x$   $X$ ) (insert  $x$   $B$ )
  then have rank-of (insert  $x$   $X$ ) = card (insert  $x$   $B$ )
    using assms rank-of-eq-card-basis-in by auto
  also have ... = Suc (card ( $B - \{x\}$ ))
    using assms card.insert-remove[of  $B$   $x$ ] using  $B$  basis-in-finite by auto
  also have ... ≤ Suc (card  $B$ )
    using assms  $B$  basis-in-finite card-Diff1-le[of  $B$ ] by auto
  also have ... = Suc (rank-of  $X$ )
    using assms  $B$  rank-of-eq-card-basis-in by auto
  finally show ?thesis .
qed
qed

lemma rank-of-Un-Int-le:
  assumes  $X \subseteq \text{carrier}$ 
  assumes  $Y \subseteq \text{carrier}$ 
  shows rank-of ( $X \cup Y$ ) + rank-of ( $X \cap Y$ ) ≤ rank-of  $X$  + rank-of  $Y$ 
proof -
  obtain  $B\text{-Int}$  where  $B\text{-Int}$ : basis-in ( $X \cap Y$ )  $B\text{-Int}$  using assms basis-in-ex[of
 $X \cap Y$ ] by auto
  then have indep-in ( $X \cup Y$ )  $B\text{-Int}$ 
    using assms indep-in-subI-subset[OF - basis-in-indep-in[of  $X \cap Y$   $B\text{-Int}$ ], of  $X
\cup Y$ ] by auto
  then obtain  $B\text{-Un}$  where  $B\text{-Un}$ : basis-in ( $X \cup Y$ )  $B\text{-Un}$   $B\text{-Int} \subseteq B\text{-Un}$ 
    using assms indep-in-imp-subset-basis-in[of  $X \cup Y$   $B\text{-Int}$ ] by auto

  have card ( $B\text{-Un} \cap (X \cup Y)$ ) + card ( $B\text{-Un} \cap (X \cap Y)$ ) = card (( $B\text{-Un} \cap X$ )
 $\cup$  ( $B\text{-Un} \cap Y$ )) + card (( $B\text{-Un} \cap X$ )  $\cap$  ( $B\text{-Un} \cap Y$ ))
  by (simp add: inf-assoc inf-left-commute inf-sup-distrib1)
  also have ... = card ( $B\text{-Un} \cap X$ ) + card ( $B\text{-Un} \cap Y$ )
proof -
  have finite ( $B\text{-Un} \cap X$ ) finite ( $B\text{-Un} \cap Y$ )
    using assms finite-subset[OF - carrier-finite] by auto
  then show ?thesis using card-Un-Int[of  $B\text{-Un} \cap X$   $B\text{-Un} \cap Y$ ] by auto
qed
also have ... ≤ rank-of  $X$  + rank-of  $Y$ 
proof -

```

**have**  $\text{card } (B\text{-Un} \cap X) \leq \text{rank-of } X$   
**proof** –  
**have**  $\text{indep-in } X (B\text{-Un} \cap X)$  **using**  $\text{assms basis-in-indep-in}[OF - B\text{-Un}(1)]$   
*indep-in-subI* **by** *auto*  
**then show** *?thesis* **using**  $\text{assms rank-of-indep-in-le}$  **by** *auto*  
**qed**  
**moreover have**  $\text{card } (B\text{-Un} \cap Y) \leq \text{rank-of } Y$   
**proof** –  
**have**  $\text{indep-in } Y (B\text{-Un} \cap Y)$  **using**  $\text{assms basis-in-indep-in}[OF - B\text{-Un}(1)]$   
*indep-in-subI* **by** *auto*  
**then show** *?thesis* **using**  $\text{assms rank-of-indep-in-le}$  **by** *auto*  
**qed**  
**ultimately show** *?thesis* **by** *auto*  
**qed**  
**finally have**  $\text{rank-of } X + \text{rank-of } Y \geq \text{card } (B\text{-Un} \cap (X \cup Y)) + \text{card } (B\text{-Un} \cap (X \cap Y))$  .  
**moreover have**  $B\text{-Un} \cap (X \cup Y) = B\text{-Un}$  **using**  $\text{assms basis-in-subset-carrier}[OF - B\text{-Un}(1)]$  **by** *auto*  
**moreover have**  $B\text{-Un} \cap (X \cap Y) = B\text{-Int}$   
**proof** –  
**have**  $\text{card } (B\text{-Un} \cap (X \cap Y)) \leq \text{card } B\text{-Int}$   
**proof** –  
**have**  $\text{indep-in } (X \cap Y) (B\text{-Un} \cap (X \cap Y))$   
**using**  $\text{assms basis-in-indep-in}[OF - B\text{-Un}(1)]$  *indep-in-subI* **by** *auto*  
**then show** *?thesis* **using**  $\text{assms basis-in-indep-in-card}[of X \cap Y - B\text{-Int}]$   
*B-Int* **by** *auto*  
**qed**  
**moreover have**  $\text{finite } (B\text{-Un} \cap (X \cap Y))$   
**using**  $\text{assms carrier-finite finite-subset}[of B\text{-Un} \cap (X \cap Y)]$  **by** *auto*  
**moreover have**  $B\text{-Int} \subseteq B\text{-Un} \cap (X \cap Y)$   
**using**  $\text{assms } B\text{-Un } B\text{-Int } \text{basis-in-subset-carrier}[of X \cap Y B\text{-Int}]$  **by** *auto*  
**ultimately show** *?thesis* **using**  $\text{card-seteq}$  **by** *blast*  
**qed**  
**ultimately have**  $\text{rank-of } X + \text{rank-of } Y \geq \text{card } B\text{-Un} + \text{card } B\text{-Int}$  **by** *auto*  
**moreover have**  $\text{card } B\text{-Un} = \text{rank-of } (X \cup Y)$   
**using**  $\text{assms rank-of-eq-card-basis-in}[OF - B\text{-Un}(1)]$  **by** *auto*  
**moreover have**  $\text{card } B\text{-Int} = \text{rank-of } (X \cap Y)$   
**using**  $\text{assms rank-of-eq-card-basis-in}[OF - B\text{-Int}]$  **by** *fastforce*  
**ultimately show**  $\text{rank-of } X + \text{rank-of } Y \geq \text{rank-of } (X \cup Y) + \text{rank-of } (X \cap Y)$  **by** *auto*  
**qed**

**lemma** *rank-of-Un-absorbI*:  
**assumes**  $X \subseteq \text{carrier } Y \subseteq \text{carrier}$   
**assumes**  $\bigwedge y. y \in Y - X \implies \text{rank-of } (\text{insert } y X) = \text{rank-of } X$   
**shows**  $\text{rank-of } (X \cup Y) = \text{rank-of } X$   
**proof** –  
**have**  $\text{finite } (Y - X)$  **using**  $\text{finite-subset}[OF \langle Y \subseteq \text{carrier} \rangle]$   $\text{carrier-finite}$  **by** *auto*

**then show** *?thesis using assms*  
**proof** (*induction*  $Y - X$  *arbitrary: Y rule: finite-induct*)  
  **case** *empty*  
    **then have**  $X \cup Y = X$  **by** *auto*  
    **then show** *?case by auto*  
  **next**  
    **case** (*insert y F*)  
    **have**  $\text{rank-of } (X \cup Y) + \text{rank-of } X \leq \text{rank-of } X + \text{rank-of } X$   
    **proof** –  
      **have**  $\text{rank-of } (X \cup Y) + \text{rank-of } X = \text{rank-of } ((X \cup (Y - \{y\})) \cup (\text{insert } y X)) + \text{rank-of } ((X \cup (Y - \{y\})) \cap (\text{insert } y X))$   
      **proof** –  
      **have**  $X \cup Y = (X \cup (Y - \{y\})) \cup (\text{insert } y X)$   $X = (X \cup (Y - \{y\})) \cap (\text{insert } y X)$  **using** *insert by auto*  
      **then show** *?thesis by auto*  
      **qed**  
      **also have**  $\dots \leq \text{rank-of } (X \cup (Y - \{y\})) + \text{rank-of } (\text{insert } y X)$   
      **proof** (*rule rank-of-Un-Int-le*)  
      **show**  $X \cup (Y - \{y\}) \subseteq \text{carrier}$  **using** *insert by auto*  
      **next**  
      **show**  $\text{insert } y X \subseteq \text{carrier}$  **using** *insert by auto*  
      **qed**  
      **also have**  $\dots = \text{rank-of } (X \cup (Y - \{y\})) + \text{rank-of } X$   
      **proof** –  
      **have**  $y \in Y - X$  **using** *insert by auto*  
      **then show** *?thesis using insert by auto*  
      **qed**  
      **also have**  $\dots = \text{rank-of } X + \text{rank-of } X$   
      **proof** –  
      **have**  $F = (Y - \{y\}) - X$   $Y - \{y\} \subseteq \text{carrier}$  **using** *insert by auto*  
      **then show** *?thesis using insert insert(3)[of Y - {y}] by auto*  
      **qed**  
      **finally show** *?thesis .*  
    **qed**  
    **moreover have**  $\text{rank-of } (X \cup Y) + \text{rank-of } X \geq \text{rank-of } X + \text{rank-of } X$   
      **using** *insert rank-of-mono by auto*  
    **ultimately show** *?case by auto*  
  **qed**  
**qed**

**lemma** *indep-iff-rank-of:*  
  **assumes**  $X \subseteq \text{carrier}$   
  **shows**  $\text{indep } X \iff \text{rank-of } X = \text{card } X$   
**proof** (*standard, goal-cases LTR RTL*)  
  **case** *LTR*  
    **then have** *indep-in X X by (auto intro: indep-inI)*  
    **then have** *basis-in X X by (auto intro: basis-inI[OF assms])*  
    **then show** *?case using rank-of-eq-card-basis-in[OF assms] by auto*  
  **next**

**case** *RTL*  
**obtain**  $B$  **where**  $B$ : *basis-in*  $X$   $B$  **using** *basis-in-ex*[*OF* *assms*] **by** *auto*  
**then have**  $\text{card } B = \text{card } X$  **using** *RTL rank-of-eq-card-basis-in*[*OF* *assms*] **by**  
*auto*  
**then have**  $B = X$   
**using** *basis-in-subset-carrier*[*OF* *assms*  $B$ ] *card-seteq*[*OF* *finite-subset*[*OF* *assms*  
*carrier-finite*]]  
**by** *auto*  
**then show** *?case* **using** *basis-in-indep-in*[*OF* *assms*  $B$ ] *indep-in-indep* **by** *auto*  
**qed**

**lemma** *basis-iff-rank-of*:

**assumes**  $X \subseteq \text{carrier}$   
**shows** *basis*  $X \iff \text{rank-of } X = \text{card } X \wedge \text{rank-of } X = \text{rank-of carrier}$   
**proof** (*standard*, *goal-cases* *LTR* *RTL*)  
**case** *LTR*  
**then have**  $\text{rank-of } X = \text{card } X$  **using** *assms indep-iff-rank-of basis-indep* **by**  
*auto*  
**moreover have**  $\dots = \text{rank-of carrier}$   
**using** *LTR rank-of-eq-card-basis-in*[*of* *carrier*  $X$ ] *basis-iff-basis-in* **by** *auto*  
**ultimately show** *?case* **by** *auto*

**next**

**case** *RTL*  
**show** *?case*  
**proof** (*rule* *basisI*)  
**show** *indep*  $X$  **using** *assms RTL indep-iff-rank-of* **by** *blast*

**next**

**fix**  $x$   
**assume**  $x \in \text{carrier} - X$   
**show**  $\neg \text{indep}$  (*insert*  $x$   $X$ )  
**proof** (*rule* *ccontr*, *goal-cases* *False*)  
**case** *False*  
**then have**  $\text{card}$  (*insert*  $x$   $X$ )  $\leq \text{rank-of carrier}$   
**using** *assms x indep-inI rank-of-indep-in-le* **by** *auto*  
**also have**  $\dots = \text{card } X$  **using** *RTL* **by** *auto*  
**finally show** *?case* **using** *finite-subset*[*OF* *assms carrier-finite*]  $x$  **by** *auto*  
**qed**

**qed**

**qed**

**lemma** *circuit-iff-rank-of*:

**assumes**  $X \subseteq \text{carrier}$   
**shows** *circuit*  $X \iff X \neq \{\} \wedge (\forall x \in X. \text{rank-of } (X - \{x\}) = \text{card } (X - \{x\})$   
 $\wedge \text{card } (X - \{x\}) = \text{rank-of } X)$   
**proof** (*standard*, *goal-cases* *LTR* *RTL*)  
**case** *LTR*  
**then have**  $X \neq \{\}$  **using** *circuit-nonempty* **by** *auto*  
**moreover have** *indep-remove*:  $\bigwedge x. x \in X \implies \text{rank-of } (X - \{x\}) = \text{card } (X -$   
 $\{x\})$

```

proof –
  fix  $x$ 
  assume  $x \in X$ 
  then have  $\text{indep } (X - \{x\})$  using  $\text{circuit-min-dep}[OF LTR]$  by auto
  moreover have  $X - \{x\} \subseteq \text{carrier}$  using  $\text{assms}$  by auto
  ultimately show  $\text{rank-of } (X - \{x\}) = \text{card } (X - \{x\})$  using  $\text{indep-iff-rank-of}$ 
by auto
qed
moreover have  $\bigwedge x. x \in X \implies \text{rank-of } (X - \{x\}) = \text{rank-of } X$ 
proof –
  fix  $x$ 
  assume  $x \in X$ 
  have  $\text{rank-of } X \leq \text{card } X$  using  $\text{assms rank-of-le}$  by auto
  moreover have  $\text{rank-of } X \neq \text{card } X$  using  $\text{assms LTR circuitD indep-iff-rank-of}[of$ 
 $X]$  by auto
  ultimately have  $\text{rank-of } X < \text{card } X$  by auto
  then have  $\text{rank-of } X \leq \text{card } (X - \{x\})$  using  $* \text{finite-subset}[OF \text{assms}]$ 
 $\text{carrier-finite}$  by auto
  also have  $\dots = \text{rank-of } (X - \{x\})$  using  $\text{indep-remove } \langle x \in X \rangle$  by auto
  finally show  $\text{rank-of } (X - \{x\}) = \text{rank-of } X$  using  $\text{assms rank-of-mono}[of X$ 
 $- \{x\} X]$  by auto
qed
ultimately show  $?case$  by auto
next
case  $RTL$ 
then have  $X \neq \{\}$ 
  and  $\text{indep-remove}: \bigwedge x. x \in X \implies \text{rank-of } (X - \{x\}) = \text{card } (X - \{x\})$ 
  and  $\text{dep}: \bigwedge x. x \in X \implies \text{rank-of } (X - \{x\}) = \text{rank-of } X$ 
  by auto
show  $?case$  using  $\text{assms}$ 
proof ( $\text{rule circuitI}$ )
  obtain  $x$  where  $x: x \in X$  using  $\langle X \neq \{\} \rangle$  by auto
  then have  $\text{rank-of } X = \text{card } (X - \{x\})$  using  $\text{dep indep-remove}$  by auto
  also have  $\dots < \text{card } X$  using  $\text{card-Diff1-less}[OF \text{finite-subset}[OF \text{assms carrier-finite}] x]$  .
  finally show  $\neg \text{indep } X$  using  $\text{indep-iff-rank-of}[OF \text{assms}]$  by auto
next
  fix  $x$ 
  assume  $x \in X$ 
  then show  $\text{indep } (X - \{x\})$  using  $\text{assms indep-remove}[of x] \text{indep-iff-rank-of}[of$ 
 $X - \{x\}]$ 
  by auto
qed
qed

context
  fixes  $\mathcal{E}$ 
  assumes  $*: \mathcal{E} \subseteq \text{carrier}$ 
begin

```

**interpretation**  $\mathcal{E}$ : *matroid*  $\mathcal{E}$  *indep-in*  $\mathcal{E}$   
**using** \* **by** *auto*

**lemma** *indep-in-iff-rank-of*:  
**assumes**  $X \subseteq \mathcal{E}$   
**shows** *indep-in*  $\mathcal{E}$   $X \longleftrightarrow \text{rank-of } X = \text{card } X$   
**using** *assms*  $\mathcal{E}$ .*indep-iff-rank-of* *rank-of-sub-cong*[*OF* \* *assms*] **by** *auto*

**lemma** *basis-in-iff-rank-of*:  
**assumes**  $X \subseteq \mathcal{E}$   
**shows** *basis-in*  $\mathcal{E}$   $X \longleftrightarrow \text{rank-of } X = \text{card } X \wedge \text{rank-of } X = \text{rank-of } \mathcal{E}$   
**using**  $\mathcal{E}$ .*basis-iff-rank-of*[*OF* *assms*] *rank-of-sub-cong*[*OF* \*] *assms*  
**unfolding** *basis-in-def* **by** *auto*

**lemma** *circuit-in-iff-rank-of*:  
**assumes**  $X \subseteq \mathcal{E}$   
**shows** *circuit-in*  $\mathcal{E}$   $X \longleftrightarrow X \neq \{\} \wedge (\forall x \in X. \text{rank-of } (X - \{x\}) = \text{card } (X - \{x\}) \wedge \text{card } (X - \{x\}) = \text{rank-of } X)$   
**proof** –  
**have** *circuit-in*  $\mathcal{E}$   $X \longleftrightarrow \mathcal{E}$ .*circuit*  $X$  **unfolding** *circuit-in-def* ..  
**also have** ...  $\longleftrightarrow X \neq \{\} \wedge (\forall x \in X. \mathcal{E}$ .*rank-of*  $(X - \{x\}) = \text{card } (X - \{x\}) \wedge \text{card } (X - \{x\}) = \mathcal{E}$ .*rank-of*  $X)$   
**using**  $\mathcal{E}$ .*circuit-iff-rank-of*[*OF* *assms*] .  
**also have** ...  $\longleftrightarrow X \neq \{\} \wedge (\forall x \in X. \text{rank-of } (X - \{x\}) = \text{card } (X - \{x\}) \wedge \text{card } (X - \{x\}) = \text{rank-of } X)$   
**proof** –  
{  
**fix**  $x$   
**have**  $\mathcal{E}$ .*rank-of*  $(X - \{x\}) = \text{rank-of } (X - \{x\})$   $\mathcal{E}$ .*rank-of*  $X = \text{rank-of } X$   
**using** *assms* *rank-of-sub-cong*[*OF* \*, *of*  $X - \{x\}$ ] *rank-of-sub-cong*[*OF* \*, *of*  $X$ ] **by** *auto*  
**then have**  $\mathcal{E}$ .*rank-of*  $(X - \{x\}) = \text{card } (X - \{x\}) \wedge \text{card } (X - \{x\}) = \mathcal{E}$ .*rank-of*  $X \longleftrightarrow \text{rank-of } (X - \{x\}) = \text{card } (X - \{x\}) \wedge \text{card } (X - \{x\}) = \text{rank-of } X$   
**by** *auto*  
}  
**then show** *?thesis*  
**by** (*auto simp: simp del: card-Diff-insert*)  
**qed**  
**finally show** *?thesis* .  
**qed**  
**end**

## 2.5 Closure

**definition** *cl* :: 'a set  $\Rightarrow$  'a set **where**  
 $cl\ X \equiv \{x \in \text{carrier}. \text{rank-of } (\text{insert } x\ X) = \text{rank-of } X\}$

**lemma** *clI*:  
**assumes**  $x \in \text{carrier}$   
**assumes**  $\text{rank-of } (\text{insert } x \ X) = \text{rank-of } X$   
**shows**  $x \in \text{cl } X$   
**unfolding** *cl-def* **using** *assms* **by** *auto*

**lemma** *cl-altdef*:  
**assumes**  $X \subseteq \text{carrier}$   
**shows**  $\text{cl } X = \bigcup \{Y \in \text{Pow } \text{carrier}. X \subseteq Y \wedge \text{rank-of } Y = \text{rank-of } X\}$   
**proof** –  
{  
  **fix**  $x$   
  **assume**  $*$ :  $x \in \text{cl } X$   
  **have**  $x \in \bigcup \{Y \in \text{Pow } \text{carrier}. X \subseteq Y \wedge \text{rank-of } Y = \text{rank-of } X\}$   
  **proof**  
    **show**  $\text{insert } x \ X \in \{Y \in \text{Pow } \text{carrier}. X \subseteq Y \wedge \text{rank-of } Y = \text{rank-of } X\}$   
    **using** *assms*  $*$  **unfolding** *cl-def* **by** *auto*  
  **qed** *auto*  
}  
**moreover** {  
  **fix**  $x$   
  **assume**  $*$ :  $x \in \bigcup \{Y \in \text{Pow } \text{carrier}. X \subseteq Y \wedge \text{rank-of } Y = \text{rank-of } X\}$   
  **then obtain**  $Y$  **where**  $Y: x \in Y \ Y \subseteq \text{carrier} \ X \subseteq Y \ \text{rank-of } Y = \text{rank-of } X$   
**by** *auto*  
  **have**  $\text{rank-of } (\text{insert } x \ X) = \text{rank-of } X$   
  **proof** –  
    **have**  $\text{rank-of } (\text{insert } x \ X) \leq \text{rank-of } X$   
    **proof** –  
      **have**  $\text{insert } x \ X \subseteq Y$  **using**  $Y$  **by** *auto*  
      **then show** *?thesis* **using** *rank-of-mono*[*of insert x X Y*]  $Y$  **by** *auto*  
    **qed**  
    **moreover have**  $\text{rank-of } X \leq \text{rank-of } (\text{insert } x \ X)$  **using**  $Y$  **by** (*auto intro*:  
*rank-of-mono*)  
    **ultimately show** *?thesis* **by** *auto*  
  **qed**  
  **then have**  $x \in \text{cl } X$  **using**  $*$  **unfolding** *cl-def* **by** *auto*  
}  
**ultimately show** *?thesis* **by** *blast*  
**qed**

**lemma** *cl-rank-of*:  $x \in \text{cl } X \implies \text{rank-of } (\text{insert } x \ X) = \text{rank-of } X$   
**unfolding** *cl-def* **by** *auto*

**lemma** *cl-subset-carrier*:  $\text{cl } X \subseteq \text{carrier}$   
**unfolding** *cl-def* **by** *auto*

**lemmas** *clD* = *cl-rank-of cl-subset-carrier*

```

lemma cl-subset:
  assumes  $X \subseteq \text{carrier}$ 
  shows  $X \subseteq \text{cl } X$ 
  using assms using insert-absorb[of - X] by (auto intro!: clI)

lemma cl-mono:
  assumes  $X \subseteq Y$ 
  assumes  $Y \subseteq \text{carrier}$ 
  shows  $\text{cl } X \subseteq \text{cl } Y$ 
proof
  fix  $x$ 
  assume  $x \in \text{cl } X$ 
  then have  $x \in \text{carrier}$  using cl-subset-carrier by auto

  have  $\text{insert } x X \subseteq \text{carrier}$ 
    using assms  $\langle x \in \text{cl } X \rangle$  cl-subset-carrier[of X] by auto
  then interpret X-insert: matroid insert x X indep-in (insert x X) by auto

  have  $\text{insert } x Y \subseteq \text{carrier}$ 
    using assms  $\langle x \in \text{cl } X \rangle$  cl-subset-carrier[of X] by auto
  then interpret Y-insert: matroid insert x Y indep-in (insert x Y) by auto

  show  $x \in \text{cl } Y$  using  $\langle x \in \text{carrier} \rangle$ 
proof (rule clI, cases  $x \in X$ )
  case True
    then show  $\text{rank-of (insert } x Y) = \text{rank-of } Y$  using assms insert-absorb[of x Y] by auto
  next
  case False
    obtain  $B_X$  where  $B_X$ : basis-in X B_X using assms basis-in-ex[of X] by auto

    have basis-in (insert x X) B_X
    proof -
      have  $\text{rank-of } B_X = \text{card } B_X \wedge \text{rank-of } B_X = \text{rank-of (insert } x X)$ 
      proof -
        have  $\text{rank-of } B_X = \text{card } B_X \wedge \text{rank-of } B_X = \text{rank-of } X$ 
          using assms B_X
          basis-in-subset-carrier[where  $\mathcal{E} = X$  and  $X = B_X$ ]
          basis-in-iff-rank-of[where  $\mathcal{E} = X$  and  $X = B_X$ ]
          by blast
        then show ?thesis using cl-rank-of[OF  $\langle x \in \text{cl } X \rangle$ ] by auto
      qed
    then show ?thesis
    using assms basis-in-subset-carrier[OF - B_X]  $\langle x \in \text{carrier} \rangle$  basis-in-iff-rank-of[of insert x X B_X]
    by auto
  qed

```

```

have indep-in (insert x Y) BX
  using assms basis-in-indep-in[OF - BX] indep-in-subI-subset[of X insert x Y]
by auto
then obtain BY where BY: basis-in (insert x Y) BY BX ⊆ BY
  using assms ⟨x ∈ carrier⟩ indep-in-iff-subset-basis-in[of insert x Y BX] by
  auto

have basis-in Y BY
proof -
  have x ∉ BY
  proof (rule ccontr, goal-cases False)
  case False
  then have insert x BX ⊆ BY using ⟨BX ⊆ BY⟩ by auto
  then have indep-in (insert x Y) (insert x BX)
    using assms ⟨x ∈ carrier⟩
    BY basis-in-indep-in[where  $\mathcal{E} = \text{insert } x \text{ Y}$  and  $X = B_Y$ ]
    indep-in-subset[where  $\mathcal{E} = \text{insert } x \text{ Y}$  and  $X = B_Y$  and  $Y = \text{insert } x$ 
BX]
  by auto
  then have indep-in (insert x X) (insert x BX)
    using assms BX
    basis-in-subset-carrier[where  $\mathcal{E} = X$  and  $X = B_X$ ]
    indep-in-supI[where  $\mathcal{E} = \text{insert } x \text{ Y}$  and  $\mathcal{E}' = \text{insert } x \text{ X}$  and  $X =$ 
insert x BX]
  by auto
  moreover have x ∈ insert x X - BX
  using assms ⟨x ∉ X⟩ BX basis-in-subset-carrier[where  $\mathcal{E} = X$  and  $X =$ 
BX] by auto
  ultimately show ?case
    using assms ⟨x ∈ carrier⟩ ⟨basis-in (insert x X) BX⟩
    basis-in-max-indep-in[where  $\mathcal{E} = \text{insert } x \text{ X}$  and  $X = B_X$  and  $x = x$ ]
    by auto
  qed
  then show ?thesis
  using assms ⟨x ∈ carrier⟩ BY basis-in-subset-carrier[of insert x Y BY]
    basis-in-supI[where  $\mathcal{E} = \text{insert } x \text{ Y}$  and  $\mathcal{E}' = Y$  and  $B = B_Y$ ] by auto
  qed

show rank-of (insert x Y) = rank-of Y
proof -
  have rank-of (insert x Y) = card BY
  using assms ⟨x ∈ carrier⟩ ⟨basis-in (insert x Y) BY⟩ basis-in-subset-carrier
  using basis-in-iff-rank-of[where  $\mathcal{E} = \text{insert } x \text{ Y}$  and  $X = B_Y$ ]
  by auto
  also have ... = rank-of Y
  using assms ⟨x ∈ carrier⟩ ⟨basis-in Y BY⟩ basis-in-subset-carrier
  using basis-in-iff-rank-of[where  $\mathcal{E} = Y$  and  $X = B_Y$ ]
  by auto
  finally show ?thesis .

```

```

    qed
  qed
qed

lemma cl-insert-absorb:
  assumes  $X \subseteq \text{carrier}$ 
  assumes  $x \in \text{cl } X$ 
  shows  $\text{cl } (\text{insert } x \ X) = \text{cl } X$ 
proof
  show  $\text{cl } (\text{insert } x \ X) \subseteq \text{cl } X$ 
  proof (standard, goal-cases elem)
    case (elem y)
    then have  $*: x \in \text{carrier } y \in \text{carrier}$  using assms cl-subset-carrier by auto

    have  $\text{rank-of } (\text{insert } y \ X) = \text{rank-of } (\text{insert } y \ (\text{insert } x \ X))$ 
    proof -
      have  $\text{rank-of } (\text{insert } y \ X) \leq \text{rank-of } (\text{insert } y \ (\text{insert } x \ X))$ 
      using assms * by (auto intro: rank-of-mono)
      moreover have  $\text{rank-of } (\text{insert } y \ (\text{insert } x \ X)) = \text{rank-of } (\text{insert } y \ X)$ 
      proof -
        have  $\text{insert } y \ (\text{insert } x \ X) = \text{insert } x \ (\text{insert } y \ X)$  by auto
        then have  $\text{rank-of } (\text{insert } y \ (\text{insert } x \ X)) = \text{rank-of } (\text{insert } x \ (\text{insert } y \ X))$ 
      by auto
      also have  $\dots = \text{rank-of } (\text{insert } y \ X)$ 
      proof -
        have  $\text{cl } X \subseteq \text{cl } (\text{insert } y \ X)$  by (rule cl-mono) (auto simp add: assms  $\langle y \in \text{carrier} \rangle$ )
        then have  $x \in \text{cl } (\text{insert } y \ X)$  using assms by auto
        then show ?thesis unfolding cl-def by auto
      qed
      finally show ?thesis .
    qed
    ultimately show ?thesis by auto
  qed
  also have  $\dots = \text{rank-of } (\text{insert } x \ X)$  using elem using cl-rank-of by auto
  also have  $\dots = \text{rank-of } X$  using assms cl-rank-of by auto
  finally show  $y \in \text{cl } X$  using * by (auto intro: clI)
qed
next
  have  $\text{insert } x \ X \subseteq \text{carrier}$  using assms cl-subset-carrier by auto
  moreover have  $X \subseteq \text{insert } x \ X$  using assms by auto
  ultimately show  $\text{cl } X \subseteq \text{cl } (\text{insert } x \ X)$  using assms cl-subset-carrier cl-mono
by auto
qed

lemma cl-cl-absorb:
  assumes  $X \subseteq \text{carrier}$ 
  shows  $\text{cl } (\text{cl } X) = \text{cl } X$ 
proof

```

```

show  $cl (cl X) \subseteq cl X$ 
proof (standard, goal-cases elem)
  case (elem x)
  then have  $x \in carrier$  using cl-subset-carrier by auto
  then show ?case
  proof (rule cI)
    have rank-of (insert x X)  $\geq$  rank-of X
      using assms  $\langle x \in carrier \rangle$  rank-of-mono[of X insert x X] by auto
    moreover have rank-of (insert x X)  $\leq$  rank-of X
    proof -
      have rank-of (insert x X)  $\leq$  rank-of (insert x (cl X))
        using assms  $\langle x \in carrier \rangle$  cl-subset-carrier cl-subset[of X]
          rank-of-mono[of insert x X insert x (cl X)] by auto
      also have ... = rank-of (cl X) using elem cl-rank-of by auto
      also have ... = rank-of (X  $\cup$  (cl X - X))
        using Diff-partition[OF cl-subset[OF assms]] by auto
      also have ... = rank-of X using  $\langle X \subseteq carrier \rangle$ 
    proof (rule rank-of-Un-absorbI)
      show  $cl X - X \subseteq carrier$  using assms cl-subset-carrier by auto
    next
      fix y
      assume  $y \in cl X - X - X$ 
      then show rank-of (insert y X) = rank-of X unfolding cl-def by auto
    qed
    finally show ?thesis .
  qed
  ultimately show rank-of (insert x X) = rank-of X by auto
qed
qed
next
show  $cl X \subseteq cl (cl X)$  using cl-subset[OF cl-subset-carrier] by auto
qed

```

lemma *cl-augment*:

```

assumes  $X \subseteq carrier$ 
assumes  $x \in carrier$ 
assumes  $y \in cl (insert x X) - cl X$ 
shows  $x \in cl (insert y X)$ 
using  $\langle x \in carrier \rangle$ 
proof (rule cI)
  have rank-of (insert y X)  $\leq$  rank-of (insert x (insert y X))
    using assms cl-subset-carrier by (auto intro: rank-of-mono)
  moreover have rank-of (insert x (insert y X))  $\leq$  rank-of (insert y X)
  proof -
    have rank-of (insert x (insert y X)) = rank-of (insert y (insert x X))
    proof -
      have insert x (insert y X) = insert y (insert x X) by auto
    then show ?thesis by auto
    qed
  qed
qed

```

**also have**  $\text{rank-of } (\text{insert } y (\text{insert } x X)) = \text{rank-of } (\text{insert } x X)$   
**using** *assms cl-def* **by** *auto*  
**also have**  $\dots \leq \text{Suc } (\text{rank-of } X)$   
**using** *assms cl-subset-carrier* **by** *(auto intro: rank-of-insert-le)*  
**also have**  $\dots = \text{rank-of } (\text{insert } y X)$   
**proof** –  
**have**  $\text{rank-of } (\text{insert } y X) \leq \text{Suc } (\text{rank-of } X)$   
**using** *assms cl-subset-carrier* **by** *(auto intro: rank-of-insert-le)*  
**moreover have**  $\text{rank-of } (\text{insert } y X) \neq \text{rank-of } X$   
**using** *assms cl-def* **by** *auto*  
**moreover have**  $\text{rank-of } X \leq \text{rank-of } (\text{insert } y X)$   
**using** *assms cl-subset-carrier* **by** *(auto intro: rank-of-mono)*  
**ultimately show** *?thesis* **by** *auto*  
**qed**  
**finally show** *?thesis* .  
**qed**  
**ultimately show**  $\text{rank-of } (\text{insert } x (\text{insert } y X)) = \text{rank-of } (\text{insert } y X)$  **by** *auto*  
**qed**

**lemma** *clI-insert*:

**assumes**  $x \in \text{carrier}$   
**assumes** *indep X*  
**assumes**  $\neg \text{indep } (\text{insert } x X)$   
**shows**  $x \in \text{cl } X$   
**using**  $\langle x \in \text{carrier} \rangle$   
**proof** (*rule clI*)  
**have**  $*$ :  $X \subseteq \text{carrier}$  **using** *assms indep-subset-carrier* **by** *auto*  
**then have**  $**$ :  $\text{insert } x X \subseteq \text{carrier}$  **using** *assms* **by** *auto*  
  
**have** *indep-in*  $(\text{insert } x X) X$  **using** *assms* **by** *(auto intro: indep-inI)*  
**then obtain**  $B$  **where**  $B$ : *basis-in*  $(\text{insert } x X) B$   $X \subseteq B$   
**using** *assms indep-in-iff-subset-basis-in[OF \*\*]* **by** *auto*  
**have**  $x \notin B$   
**proof** (*rule ccontr, goal-cases False*)  
**case** *False*  
**then have** *indep-in*  $(\text{insert } x X) (\text{insert } x X)$   
**using**  $B$  *indep-in-subset*[*OF \*\** *basis-in-indep-in*[*OF \*\**]] **by** *auto*  
**then show** *?case* **using** *assms indep-in-indep* **by** *auto*  
**qed**

**have** *basis-in*  $X B$  **using**  $*$

**proof** (*rule basis-inI, goal-cases indep max-indep*)  
**case** *indep*  
**show** *?case*  
**proof** (*rule indep-in-supI*[**where**  $\mathcal{E} = \text{insert } x X$ ])  
**show**  $B \subseteq X$  **using**  $B$  *basis-in-subset-carrier*[*OF \*\**]  $\langle x \notin B \rangle$  **by** *auto*  
**next**  
**show** *indep-in*  $(\text{insert } x X) B$  **using** *basis-in-indep-in*[*OF \*\**  $B(1)$ ] .  
**qed** *auto*

```

next
  case (max-indep y)
  then have  $\neg$  indep-in (insert x X) (insert y B)
    using B basis-in-max-indep-in[OF **] by auto
  then show ?case by (auto intro: indep-in-subI-subset)
qed
then show rank-of (insert x X) = rank-of X
  using B rank-of-eq-card-basis-in[OF *] rank-of-eq-card-basis-in[OF **] by auto
qed

```

```

lemma indep-in-carrier [simp]: indep-in carrier = indep
  using indep-subset-carrier by (auto simp: indep-in-def fun-eq-iff)

```

```

context
  fixes I
  defines I  $\equiv$  ( $\lambda X. X \subseteq$  carrier  $\wedge$  ( $\forall x \in X. x \notin$  cl ( $X - \{x\}$ )))
begin

```

```

lemma I-mono: I Y if  $Y \subseteq X$  I X for X Y :: 'a set
proof -
  have  $\forall x \in Y. x \notin$  cl ( $Y - \{x\}$ )
  proof (intro ballI)
    fix x assume x:  $x \in Y$ 
    with that have cl ( $Y - \{x\}$ )  $\subseteq$  cl ( $X - \{x\}$ )
      by (intro cl-mono) (auto simp: I-def)
    with that and x show  $x \notin$  cl ( $Y - \{x\}$ ) by (auto simp: I-def)
  qed
  with that show ?thesis by (auto simp: I-def)
qed

```

```

lemma clI':
  assumes I X  $x \in$  carrier  $\neg$ I (insert x X)
  shows  $x \in$  cl X
proof -
  from assms have  $x \notin X$  by (auto simp: insert-absorb)
  from assms obtain y where  $y: y \in$  insert x X  $y \in$  cl (insert x X - {y})
    by (force simp: I-def)
  show  $x \in$  cl X
  proof (cases  $x = y$ )
    case True
      thus ?thesis using assms x y by (auto simp: I-def)
    next
      case False
        have  $y \in$  cl (insert x X - {y}) by fact
        also from False have insert x X - {y} = insert x (X - {y}) by auto
        finally have  $y \in$  cl (insert x (X - {y})) - cl (X - {y})
          using assms False y unfolding I-def by blast
        hence  $x \in$  cl (insert y (X - {y}))
          using cl-augment[of X - {y} x y] assms False y by (auto simp: I-def)

```

**also from  $y$  and  $False$  have  $insert\ y\ (X - \{y\}) = X$  by *auto***  
**finally show *?thesis* .**  
**qed**  
**qed**

**lemma *matroid-I: matroid carrier I***  
**proof (*unfold-locales, goal-cases*)**  
**show *finite carrier* by (*rule carrier-finite*)**  
**next**  
**case (*4 X Y*)**  
**have  $\forall x \in Y. x \notin cl\ (Y - \{x\})$**   
**proof (*intro ballI*)**  
**fix  $x$  assume  $x: x \in Y$**   
**with *4* have  $cl\ (Y - \{x\}) \subseteq cl\ (X - \{x\})$**   
**by (*intro cl-mono*) (*auto simp: I-def*)**  
**with *4* and  $x$  show  $x \notin cl\ (Y - \{x\})$  by (*auto simp: I-def*)**  
**qed**  
**with *4* show *?case* by (*auto simp: I-def*)**  
**next**  
**case (*5 X Y*)**  
**have  $\sim(\exists X\ Y. I\ X \wedge I\ Y \wedge card\ X < card\ Y \wedge (\forall x \in Y - X. \neg I\ (insert\ x\ X)))$**   
**proof**  
**assume  $*$ :  $\exists X\ Y. I\ X \wedge I\ Y \wedge card\ X < card\ Y \wedge (\forall x \in Y - X. \neg I\ (insert\ x\ X))$**   
**(*is*  $\exists X\ Y. ?P\ X\ Y$ )**  
**define  $n$  where  $n = Max\ ((\lambda(X, Y). card\ (X \cap Y))\ '\{(X, Y). ?P\ X\ Y\})$**   
**have  $\{(X, Y). ?P\ X\ Y\} \subseteq Pow\ carrier \times Pow\ carrier$**   
**by (*auto simp: I-def*)**  
**hence *finite*:  $finite\ \{(X, Y). ?P\ X\ Y\}$**   
**by (*rule finite-subset*) (*insert carrier-finite, auto*)**  
**hence  $n \in ((\lambda(X, Y). card\ (X \cap Y))\ '\{(X, Y). ?P\ X\ Y\})$**   
**unfolding *n-def* using  $*$  by (*intro Max-in finite-imageI*) *auto***  
**then obtain  $X\ Y$  where  $XY: ?P\ X\ Y\ n = card\ (X \cap Y)$**   
**by *auto***  
**hence *finite'*:  $finite\ X\ finite\ Y$**   
**using *finite-subset*[*OF - carrier-finite*]  $XY$  by (*auto simp: I-def*)**  
**from  $XY\ finite'$  have  $\sim(Y \subseteq X)$**   
**using *card-mono*[*of X Y*] by *auto***  
**then obtain  $y$  where  $y: y \in Y - X$  by *blast***

**have *False***  
**proof (*cases*  $X \subseteq cl\ (Y - \{y\})$ )**  
**case *True***  
**from  $y\ XY$  have [*simp*]:  $y \in carrier$  by (*auto simp: I-def*)**  
**assume  $X \subseteq cl\ (Y - \{y\})$**   
**hence  $cl\ X \subseteq cl\ (cl\ (Y - \{y\}))$**   
**by (*intro cl-mono cl-subset-carrier*)**  
**also have  $\dots = cl\ (Y - \{y\})$**   
**using  $XY$  by (*intro cl-cl-absorb*) (*auto simp: I-def*)**

**finally have**  $cl\ X \subseteq cl\ (Y - \{y\})$  .  
**moreover have**  $y \notin cl\ (Y - \{y\})$   
**using**  $y\ I\text{-def}\ XY(1)$  **by** *blast*  
**ultimately have**  $y \notin cl\ X$  **by** *blast*  
**thus** *False* **unfolding** *I-def*  
**using**  $XY\ y\ clI'\ \langle y \in carrier \rangle$  **by** *blast*  
**next**  
**case** *False*  
**with**  $y\ XY$  **have**  $[simp]: y \in carrier$  **by** (*auto simp: I-def*)  
**assume**  $\neg(X \subseteq cl\ (Y - \{y\}))$   
**then obtain**  $t$  **where**  $t: t \in X\ t \notin cl\ (Y - \{y\})$   
**by** *auto*  
**with**  $XY$  **have**  $[simp]: t \in carrier$  **by** (*auto simp: I-def*)  
  
**have**  $t \in X - Y$   
**using**  $t\ y\ clI'[of\ t\ Y - \{y\}]$  **by** (*cases\ t = y*) (*auto simp: insert-absorb*)  
**moreover have**  $I\ (Y - \{y\})$  **using**  $XY(1)\ I\text{-mono}[of\ Y - \{y\}\ Y]$  **by** *blast*  
**ultimately have**  $*$ :  $I\ (insert\ t\ (Y - \{y\}))$   
**using**  $clI'[of\ Y - \{y\}\ t]\ t$  **by** *auto*  
  
**from**  $XY$  **have** *finite*  $Y$   
**by** (*intro\ finite-subset[OF - carrier-finite]*) (*auto simp: I-def*)  
**moreover from**  $y$  **have**  $Y \neq \{\}$  **by** *auto*  
**ultimately have**  $[simp]: card\ (insert\ t\ (Y - \{y\})) = card\ Y$  **using**  $\langle t \in X - Y \rangle\ y$   
**by** (*simp\ add: Suc-diff-Suc\ card-gt-0-iff*)  
  
**have**  $\exists x \in Y - X.\ I\ (insert\ x\ X)$   
**proof** (*rule\ ccontr*)  
**assume**  $\neg?$ *thesis*  
**hence**  $?P\ X\ (insert\ t\ (Y - \{y\}))$  **using**  $XY\ *\ \langle t \in X - Y \rangle$   
**by** *auto*  
**hence**  $card\ (X \cap insert\ t\ (Y - \{y\})) \leq n$   
**unfolding** *n-def* **using** *finite* **by** (*intro\ Max-ge*) *auto*  
**also have**  $X \cap insert\ t\ (Y - \{y\}) = insert\ t\ ((X \cap Y) - \{y\})$   
**using**  $y\ \langle t \in X - Y \rangle$  **by** *blast*  
**also have**  $card\ \dots = Suc\ (card\ (X \cap Y))$   
**using**  $y\ \langle t \in X - Y \rangle\ \langle finite\ Y \rangle$  **by** (*simp\ add: card.insert-remove*)  
**finally show** *False* **using**  $XY$  **by** *simp*  
**qed**  
**with**  $XY$  **show** *False* **by** *blast*  
**qed**  
**thus** *False* .  
**qed**  
**with**  $5$  **show** *?case* **by** *auto*  
**qed** (*auto simp: I-def*)  
  
**end**

**definition** *cl-in* where  $cl\text{-in } \mathcal{E} X = \text{matroid.cl } \mathcal{E} (\text{indep-in } \mathcal{E}) X$

**lemma** *cl-eq-cl-in*:

assumes  $X \subseteq \text{carrier}$

shows  $cl X = cl\text{-in carrier } X$

**proof** –

interpret  $\mathcal{E}$ : *matroid carrier indep-in carrier*

by (*intro matroid-subset*) *auto*

have  $cl X = \{x \in \text{carrier}. \text{rank-of } (\text{insert } x X) = \text{rank-of } X\}$

unfolding *cl-def* by *auto*

also have  $\dots = \{x \in \text{carrier}. \mathcal{E}.\text{rank-of } (\text{insert } x X) = \mathcal{E}.\text{rank-of } X\}$

using *rank-of-sub-cong[of carrier] assms* by *auto*

also have  $\dots = cl\text{-in carrier } X$

unfolding *cl-in-def*  $\mathcal{E}.\text{cl-def}$  by *auto*

finally show *?thesis* .

qed

**context**

fixes  $\mathcal{E}$

assumes  $*$ :  $\mathcal{E} \subseteq \text{carrier}$

**begin**

**interpretation**  $\mathcal{E}$ : *matroid  $\mathcal{E}$  indep-in  $\mathcal{E}$*

using  $*$  by *auto*

**lemma** *cl-inI-aux*:  $x \in \mathcal{E}.cl X \implies x \in cl\text{-in } \mathcal{E} X$

unfolding *cl-in-def* by *auto*

**lemma** *cl-inD-aux*:  $x \in cl\text{-in } \mathcal{E} X \implies x \in \mathcal{E}.cl X$

unfolding *cl-in-def* by *auto*

**lemma** *cl-inI*:

assumes  $X \subseteq \mathcal{E}$

assumes  $x \in \mathcal{E}$

assumes  $\text{rank-of } (\text{insert } x X) = \text{rank-of } X$

shows  $x \in cl\text{-in } \mathcal{E} X$

**proof** –

have  $\mathcal{E}.\text{rank-of } (\text{insert } x X) = \text{rank-of } (\text{insert } x X) \mathcal{E}.\text{rank-of } X = \text{rank-of } X$

using *assms rank-of-sub-cong[OF \*]* by *auto*

then show *?thesis* unfolding *cl-in-def* using *assms* by (*auto intro:  $\mathcal{E}.clI$* )

qed

**lemma** *cl-in-altdef*:

assumes  $X \subseteq \mathcal{E}$

shows  $cl\text{-in } \mathcal{E} X = \bigcup \{Y \in \text{Pow } \mathcal{E}. X \subseteq Y \wedge \text{rank-of } Y = \text{rank-of } X\}$

unfolding *cl-in-def*

**proof** (*safe, goal-cases LTR RTL*)

case (*LTR*  $x$ )

then have  $x \in \bigcup \{Y \in \text{Pow } \mathcal{E}. X \subseteq Y \wedge \mathcal{E}.\text{rank-of } Y = \mathcal{E}.\text{rank-of } X\}$

using  $\mathcal{E}.cl\text{-altdef}[OF\ assms]$  by auto  
 then obtain  $Y$  where  $Y: x \in Y \ Y \in Pow\ \mathcal{E} \ X \subseteq Y \ \mathcal{E}.rank\text{-of}\ Y = \mathcal{E}.rank\text{-of}\ X$  by auto  
 then show ?case using rank-of-sub-cong[ $OF\ *$ ] by auto  
 next  
 case ( $RTL\ x\ Y$ )  
 then have  $x \in \bigcup \{Y \in Pow\ \mathcal{E}. \ X \subseteq Y \wedge \mathcal{E}.rank\text{-of}\ Y = \mathcal{E}.rank\text{-of}\ X\}$   
 using rank-of-sub-cong[ $OF\ *,\ of\ X$ ] rank-of-sub-cong[ $OF\ *,\ of\ Y$ ] by auto  
 then show ?case using  $\mathcal{E}.cl\text{-altdef}[OF\ assms]$  by auto  
 qed

lemma *cl-in-subset-carrier*:  $cl\text{-in}\ \mathcal{E} \ X \subseteq \mathcal{E}$   
 using  $\mathcal{E}.cl\text{-subset-carrier}$  unfolding *cl-in-def* .

lemma *cl-in-rank-of*:  
 assumes  $X \subseteq \mathcal{E}$   
 assumes  $x \in cl\text{-in}\ \mathcal{E} \ X$   
 shows  $rank\text{-of}\ (insert\ x\ X) = rank\text{-of}\ X$   
 proof –  
 have  $\mathcal{E}.rank\text{-of}\ (insert\ x\ X) = \mathcal{E}.rank\text{-of}\ X$   
 using *assms*  $\mathcal{E}.cl\text{-rank-of}$  unfolding *cl-in-def* by auto  
 moreover have  $\mathcal{E}.rank\text{-of}\ (insert\ x\ X) = rank\text{-of}\ (insert\ x\ X)$   
 using *assms* rank-of-sub-cong[ $OF\ *,\ of\ insert\ x\ X$ ] *cl-in-subset-carrier* by auto  
 moreover have  $\mathcal{E}.rank\text{-of}\ X = rank\text{-of}\ X$   
 using *assms* rank-of-sub-cong[ $OF\ *$ ] by auto  
 ultimately show ?thesis by auto  
 qed

lemmas  $cl\text{-in}D = cl\text{-in-rank-of}\ cl\text{-in-subset-carrier}$

lemma *cl-in-subset*:  
 assumes  $X \subseteq \mathcal{E}$   
 shows  $X \subseteq cl\text{-in}\ \mathcal{E} \ X$   
 using  $\mathcal{E}.cl\text{-subset}[OF\ assms]$  unfolding *cl-in-def* .

lemma *cl-in-mono*:  
 assumes  $X \subseteq Y$   
 assumes  $Y \subseteq \mathcal{E}$   
 shows  $cl\text{-in}\ \mathcal{E} \ X \subseteq cl\text{-in}\ \mathcal{E} \ Y$   
 using  $\mathcal{E}.cl\text{-mono}[OF\ assms]$  unfolding *cl-in-def* .

lemma *cl-in-insert-absorb*:  
 assumes  $X \subseteq \mathcal{E}$   
 assumes  $x \in cl\text{-in}\ \mathcal{E} \ X$   
 shows  $cl\text{-in}\ \mathcal{E} \ (insert\ x\ X) = cl\text{-in}\ \mathcal{E} \ X$   
 using *assms*  $\mathcal{E}.cl\text{-insert-absorb}$  unfolding *cl-in-def* by auto

lemma *cl-in-augment*:  
 assumes  $X \subseteq \mathcal{E}$

**assumes**  $x \in \mathcal{E}$   
**assumes**  $y \in \text{cl-in } \mathcal{E} (\text{insert } x X) - \text{cl-in } \mathcal{E} X$   
**shows**  $x \in \text{cl-in } \mathcal{E} (\text{insert } y X)$   
**using** *assms*  $\mathcal{E}.\text{cl-augment}$  **unfolding** *cl-in-def* **by** *auto*

**lemmas**  $\text{cl-inI-insert} = \text{cl-inI-aux}[OF \ \mathcal{E}.\text{clI-insert}]$

**end**

**lemma** *cl-in-subI*:  
**assumes**  $X \subseteq \mathcal{E}' \ \mathcal{E}' \subseteq \mathcal{E} \ \mathcal{E} \subseteq \text{carrier}$   
**shows**  $\text{cl-in } \mathcal{E}' X \subseteq \text{cl-in } \mathcal{E} X$   
**proof** (*safe*, *goal-cases elem*)  
**case** (*elem x*)  
**then have**  $x \in \mathcal{E}' \ \text{rank-of } (\text{insert } x X) = \text{rank-of } X$   
**using** *assms cl-inD*[**where**  $\mathcal{E} = \mathcal{E}'$  **and**  $X = X$ ] **by** *auto*  
**then show**  $x \in \text{cl-in } \mathcal{E} X$  **using** *assms* **by** (*auto intro: cl-inI*)  
**qed**

**context**  
**fixes**  $\mathcal{E}$   
**assumes**  $*$ :  $\mathcal{E} \subseteq \text{carrier}$   
**begin**

**interpretation**  $\mathcal{E}$ : *matroid*  $\mathcal{E}$  *indep-in*  $\mathcal{E}$   
**using**  $*$  **by** *auto*

**lemma** *cl-in-sub-cong*:  
**assumes**  $X \subseteq \mathcal{E}' \ \mathcal{E}' \subseteq \mathcal{E}$   
**shows**  $\mathcal{E}.\text{cl-in } \mathcal{E}' X = \text{cl-in } \mathcal{E}' X$   
**proof** (*safe*, *goal-cases LTR RTL*)  
**case** (*LTR x*)  
**then have**  $x \in \mathcal{E}' \ \mathcal{E}.\text{rank-of } (\text{insert } x X) = \mathcal{E}.\text{rank-of } X$   
**using** *assms*  
 $\mathcal{E}.\text{cl-in-rank-of}$ [**where**  $\mathcal{E} = \mathcal{E}'$  **and**  $X = X$  **and**  $x = x$ ]  
 $\mathcal{E}.\text{cl-in-subset-carrier}$ [**where**  $\mathcal{E} = \mathcal{E}'$ ]  
**by** *auto*  
**moreover have**  $\mathcal{E}.\text{rank-of } X = \text{rank-of } X$   
**using** *assms rank-of-sub-cong*[*OF*  $*$ ] **by** *auto*  
**moreover have**  $\mathcal{E}.\text{rank-of } (\text{insert } x X) = \text{rank-of } (\text{insert } x X)$   
**using** *assms rank-of-sub-cong*[*OF*  $*$ , *of insert x X*]  $\langle x \in \mathcal{E}' \rangle$  **by** *auto*  
**ultimately show** *?case* **using** *assms*  $*$  **by** (*auto intro: cl-inI*)  
**next**  
**case** (*RTL x*)  
**then have**  $x \in \mathcal{E}' \ \text{rank-of } (\text{insert } x X) = \text{rank-of } X$   
**using**  $*$  *assms cl-inD*[**where**  $\mathcal{E} = \mathcal{E}'$  **and**  $X = X$ ] **by** *auto*  
**moreover have**  $\mathcal{E}.\text{rank-of } X = \text{rank-of } X$   
**using** *assms rank-of-sub-cong*[*OF*  $*$ ] **by** *auto*  
**moreover have**  $\mathcal{E}.\text{rank-of } (\text{insert } x X) = \text{rank-of } (\text{insert } x X)$

```
    using assms rank-of-sub-cong[OF *, of insert x X] ⟨x ∈ E'⟩ by auto
ultimately show ?case using assms by (auto intro: E.cl-inI)
qed

end
end
end
```

## References

- [1] J. Oxley. What is a matroid?, 2003.