Conservative groupoids recognize only regular languages

Martin Beaudry[1] Danny Dubé[2] Maxime Dubé[2]
Mario Latendresse Pascal Tesson[2]

[1]University of Sherbrooke

[2]Laval University (Quebec City)

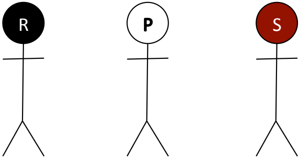
A guide for dishonest organizers of Rock-Paper-Scissors tournaments

Martin Beaudry[1] Danny Dubé[2] Maxime Dubé[2]
Mario Latendresse Pascal Tesson[2]

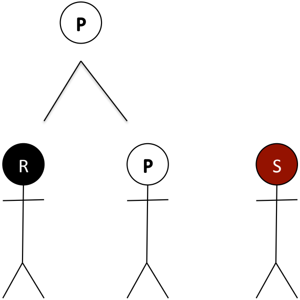
[1]University of Sherbrooke

[2]Laval University (Quebec City)

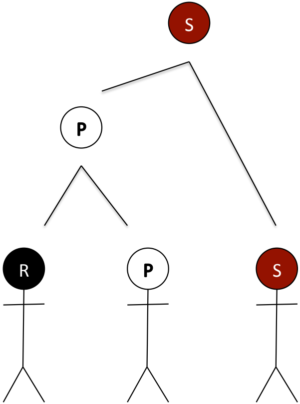
How to rig Rock-Paper-Scissors tournaments



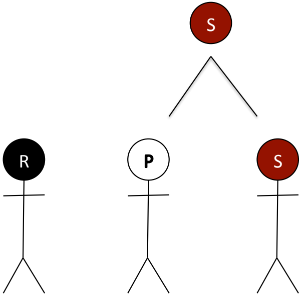
How to rig Rock-Paper-Scissors tournaments



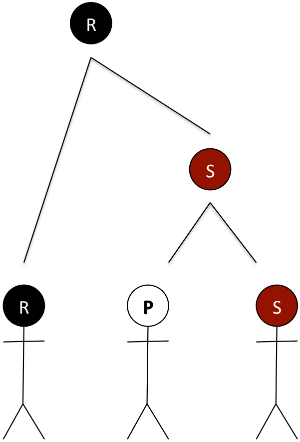
How to rig Rock-Paper-Scissors tournaments



How to rig Rock-Paper-Scissors tournaments

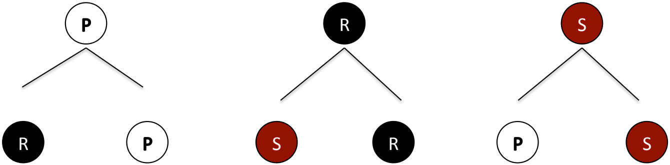


How to rig Rock-Paper-Scissors tournaments

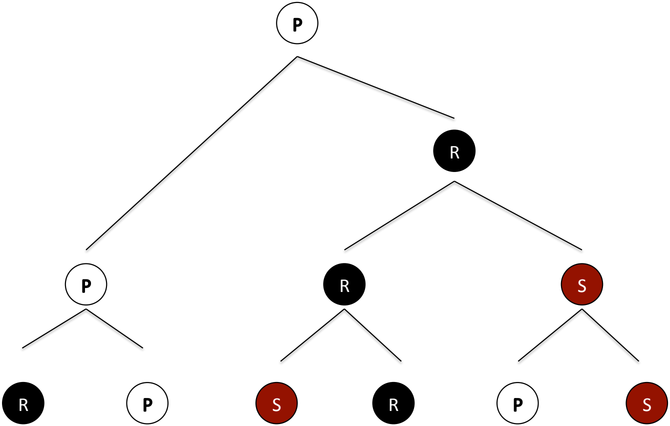


How to rig Rock-Paper-Scissors tournaments

How to rig Rock-Paper-Scissors tournaments



How to rig Rock-Paper-Scissors tournaments



Examples

In each of the following strings, is it possible to organize the tournament such that Paper is the winner?

1. P R R P R R P P R
2. R S R S S R P S
3. R S R R P R S P S R P P S P S

Examples

In each of the following strings, is it possible to organize the tournament such that Paper is the winner?

1. P R R P R R P P R
2. R S R S S R P S
3. R S R R P R S P S R P P S P S

Let $\Lambda(P) \subseteq \{R, P, S\}^*$ be the set of strings which can be evaluated to P given the right evaluation tree. How can one characterize $\Lambda(P)$?

Monoids and groupoids

Definition

A groupoid is a set with a binary operation.

Definition

A monoid is a set with a binary associative operation and an identity element for that operation.

Monoids and groupoids

Definition

A groupoid is a set with a binary operation.

Definition

A monoid is a set with a binary associative operation and an identity element for that operation.

Example

Let $H = \{R, P, S\}$. The Rock-Paper-Scissor game defines a groupoid on this set with multiplication given by

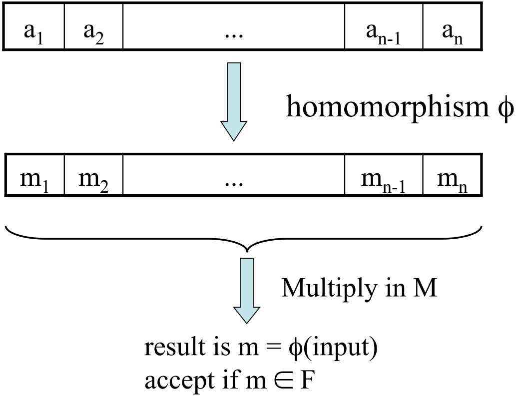
$$PP = RP = PR = P$$
$$RR = RS = SR = R$$
$$SS = PS = SP = S$$

Finite monoid ≈ finite automaton

Definition

$L \subseteq \Sigma^*$ is recognized by a finite monoid M if there exists $\phi : \Sigma \to M$ (extends to a homomorphism from Σ^* to M^*) s.t.

$x \in L \Leftrightarrow$ product $\phi(x)$ lies in some accepting subset F.



Finite monoid $\approx$ finite automaton

Definition

$L \subseteq \Sigma^*$ is recognized by a finite monoid M if there exists $\phi : \Sigma \to M$ (extends to a homomorphism from Σ^* to M^*) s.t.

$$x \in L \Leftrightarrow \text{product } \phi(x) \text{ lies in some accepting subset } F.$$

Theorem (Kleene, algebraic formulation)

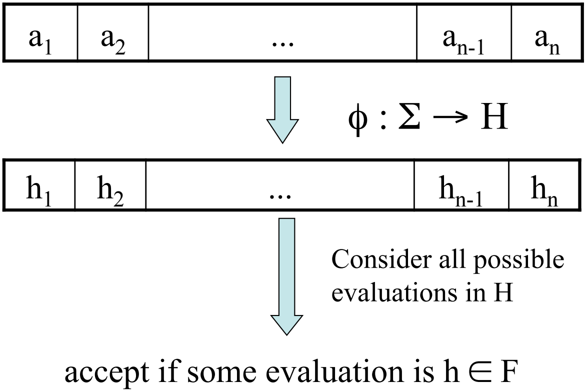
L is recognizable by a finite monoid iff L is regular.

Recognition by groupoids

Definition

$L \subseteq \Sigma^*$ is recognized by a finite groupoid H if there exists $\phi : \Sigma \to H$ s.t.

$$x \in L \Leftrightarrow \phi(x) \text{ \underline{can be evaluated} to some element in } F$$

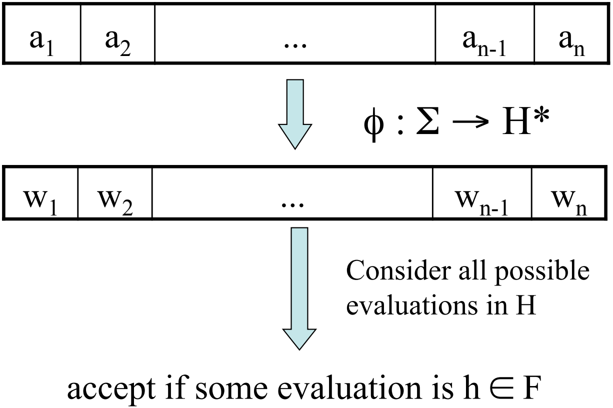


Recognition by groupoids

Definition

$L \subseteq \Sigma^*$ is recognized by a finite groupoid H if there exists $\phi : \Sigma \to H^*$ s.t.

$$x \in L \Leftrightarrow \phi(x) \text{ \underline{can be evaluated} to some element in } F$$



Recognition by groupoids

Definition

$L \subseteq \Sigma^*$ is recognized by a finite groupoid H if there exists $\phi : \Sigma \to H$ s.t.

$$x \in L \Leftrightarrow \phi(x) \text{ \underline{can be evaluated} to some element in } F$$

Theorem

L is recognizable by a finite groupoid iff L is context-free.

Groupoids without context-free capabilities

Theorem (Caussinus, Lemieux (94) - Beaudry, Lemieux, Thérien (97))

- *If H is a loop, i.e. a groupoid with an identity element and left/right inverses then H can only recognize regular languages.*
- *L is recognizable by a loop iff L is a regular open language.*

Groupoids without context-free capabilities

Theorem (Caussinus, Lemieux (94) - Beaudry, Lemieux, Thérien (97))

- *If H is a loop, i.e. a groupoid with an identity element and left/right inverses then H can only recognize regular languages.*
- *L is recognizable by a loop iff L is a regular open language.*

Theorem (Beaudry, Lemieux, Thérien (05))

If the multiplication monoid of H satisfies the identity

$$(xy)^\omega(yx)^\omega(xy)^\omega = (xy)^\omega$$

for some ω then H can only recognize regular languages.

Main result

Definition

A groupoid H is *conservative* if $xy \in \{x, y\}$ for any $x, y \in H$.

Example

- $\{r, p, s\}$ is a conservative groupoid.
- Any variant of the game with more objects (e.g. Spock, Lizard) defines a conservative (and commutative) groupoid.

Main result

Definition

A groupoid H is *conservative* if $xy \in \{x, y\}$ for any $x, y \in H$.

Example

- $\{r, p, s\}$ is a conservative groupoid.
- Any variant of the game with more objects (e.g. Spock, Lizard) defines a conservative (and commutative) groupoid.

Theorem

Any language recognized by a conservative groupoid H is regular.

Basic definitions and notation

Definition

Let H be a groupoid. Let $a \in H$. Let $\sigma \subseteq H$. Let $x \in H^*$.

- $W(x) = \{a \in H : a \text{ wins on } x \text{ given the right evaluation tree}\}$.
- $\Lambda(a) = \{x : a \in W(x)\}$.
- $\Lambda(\sigma) = \{x : W(x) \cap \sigma \neq \emptyset\} = \bigcup_{a \in \sigma} \Lambda(a)$
- $\Lambda^{\epsilon}(\sigma) = \Lambda(\sigma) \cup \{\epsilon\}$

Definition

Let H be a (commutative) conservative groupoid. An element b is *favorable* to an element a if $ba = ab = a$. We define $f(a) = \{b : b \text{ is favorable to } a\}$

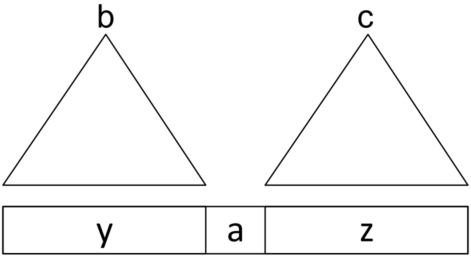
Characterizing words on which a can win

Lemma

For any $a \in H$ and any $x \in H^*$ we have $a \in W(x)$ iff $x = yaz$ such that

- *there exists $b \in W(y)$ with $b \in f(a)$ (or $y = \epsilon$)*
- *there exists $c \in W(z)$ with $c \in f(a)$ (or $z = \epsilon$)*

Equivalently $\Lambda(a) = \Lambda^\epsilon(f(a)) \cdot \{a\} \cdot \Lambda^\epsilon(f(a))$



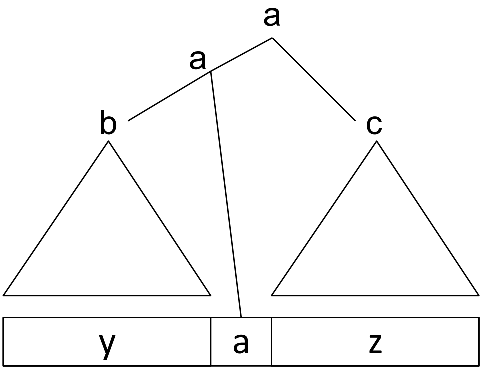
Characterizing words on which a can win

Lemma

For any $a \in H$ and any $x \in H^*$ we have $a \in W(x)$ iff $x = yaz$ such that

- *there exists $b \in W(y)$ with $b \in f(a)$ (or $y = \epsilon$)*
- *there exists $c \in W(z)$ with $c \in f(a)$ (or $z = \epsilon$)*

Equivalently $\Lambda(a) = \Lambda^{\epsilon}(f(a)) \cdot \{a\} \cdot \Lambda^{\epsilon}(f(a))$



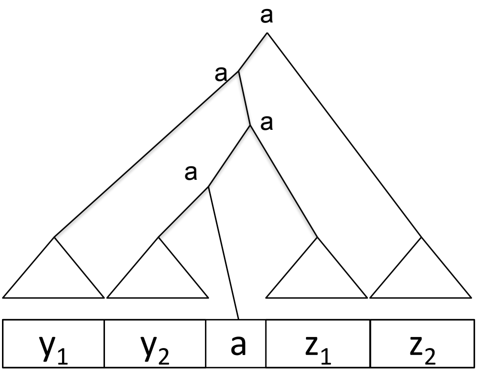
Characterizing words on which a can win

Lemma

For any $a \in H$ and any $x \in H^*$ we have $a \in W(x)$ iff $x = yaz$ such that

- *there exists $b \in W(y)$ with $b \in f(a)$ (or $y = \epsilon$)*
- *there exists $c \in W(z)$ with $c \in f(a)$ (or $z = \epsilon$)*

Equivalently $\Lambda(a) = \Lambda^\epsilon(f(a)) \cdot \{a\} \cdot \Lambda^\epsilon(f(a))$



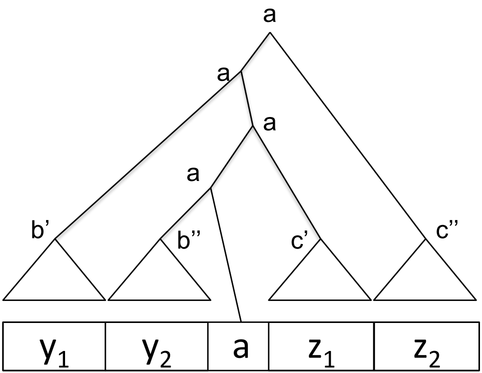
Characterizing words on which a can win

Lemma

For any $a \in H$ and any $x \in H^*$ we have $a \in W(x)$ iff $x = yaz$ such that

- *there exists $b \in W(y)$ with $b \in f(a)$ (or $y = \epsilon$)*
- *there exists $c \in W(z)$ with $c \in f(a)$ (or $z = \epsilon$)*

Equivalently $\Lambda(a) = \Lambda^\epsilon(f(a)) \cdot \{a\} \cdot \Lambda^\epsilon(f(a))$



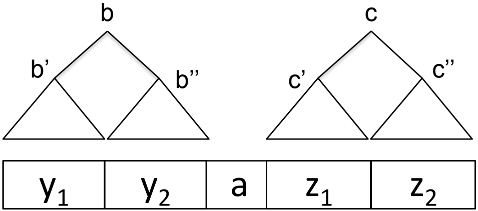
Characterizing words on which a can win

Lemma

For any $a \in H$ and any $x \in H^*$ we have $a \in W(x)$ iff $x = yaz$ such that

- *there exists $b \in W(y)$ with $b \in f(a)$ (or $y = \epsilon$)*
- *there exists $c \in W(z)$ with $c \in f(a)$ (or $z = \epsilon$)*

Equivalently $\Lambda(a) = \Lambda^{\epsilon}(f(a)) \cdot \{a\} \cdot \Lambda^{\epsilon}(f(a))$



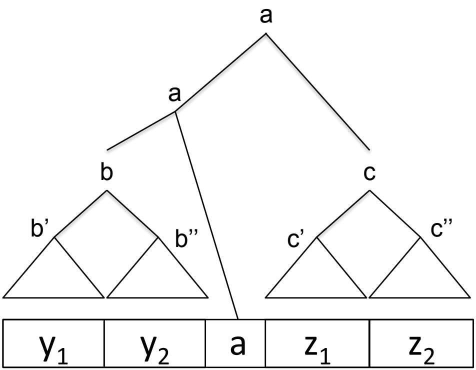
Characterizing words on which a can win

Lemma

For any $a \in H$ and any $x \in H^*$ we have $a \in W(x)$ iff $x = yaz$ such that

- *there exists $b \in W(y)$ with $b \in f(a)$ (or $y = \epsilon$)*
- *there exists $c \in W(z)$ with $c \in f(a)$ (or $z = \epsilon$)*

Equivalently $\Lambda(a) = \Lambda^{\epsilon}(f(a)) \cdot \{a\} \cdot \Lambda^{\epsilon}(f(a))$



Characterizing words on which a can win

Lemma

For any $a \in H$ and any $x \in H^*$ we have $a \in W(x)$ iff $x = yaz$ such that

- *there exists $b \in W(y)$ with $b \in f(a)$ (or $y = \epsilon$)*
- *there exists $c \in W(z)$ with $c \in f(a)$ (or $z = \epsilon$)*

Equivalently $\Lambda(a) = \Lambda^{\epsilon}(f(a)) \cdot \{a\} \cdot \Lambda^{\epsilon}(f(a))$

Example

For Rock-Paper-Scissors $\Lambda(p) = \Lambda^{\epsilon}(\{r,p\}) \cdot \{p\} \cdot \Lambda^{\epsilon}(\{r,p\})$.

Characterizing words on which one of σ can win

Lemma

For any $\sigma \subseteq H$

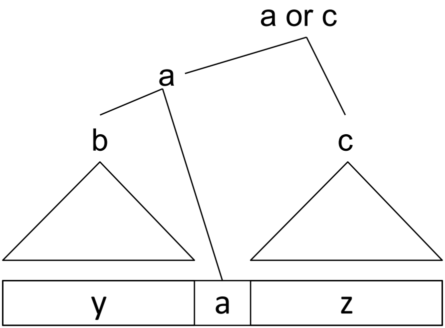
$$\Lambda(\sigma) = \bigcup_{a \in \sigma} \Lambda^{\epsilon}(f(a) \cup \sigma) \cdot \{a\} \cdot \Lambda^{\epsilon}(f(a) \cup \sigma).$$

Characterizing words on which one of σ can win

Lemma

For any $\sigma \subseteq H$

$$\Lambda(\sigma) = \bigcup_{a \in \sigma} \Lambda^{\epsilon}(f(a) \cup \sigma) \cdot \{a\} \cdot \Lambda^{\epsilon}(f(a) \cup \sigma).$$



$b \in f(a)$

$c \in \sigma$

Characterizing words on which one of σ can win

Lemma

For any $\sigma \subseteq H$

$$\Lambda(\sigma) = \bigcup_{a \in \sigma} \Lambda^{\epsilon}(f(a) \cup \sigma) \cdot \{a\} \cdot \Lambda^{\epsilon}(f(a) \cup \sigma).$$

Example

For Rock-Paper-Scissors

$\Lambda(\{p, r\}) = \Lambda^{\epsilon}(\{r, p\}) \cdot \{p\} \cdot \Lambda^{\epsilon}(\{r, p\}) \cup \Lambda^{\epsilon}(\{r, p, s\}) \cdot \{r\} \cdot \Lambda^{\epsilon}(\{r, p, s\})$

A context-free grammar for $\Lambda(a)$

Consider the following context free grammar. Non terminals are B_σ for each $\emptyset \neq \sigma \subseteq H$.

$$B_\sigma \rightarrow \epsilon \qquad \text{for all } \sigma$$

$$B_\sigma \rightarrow B_{\sigma'} a B_{\sigma'} \qquad \text{for all } a \in \sigma \text{ and } \sigma' = \sigma \cup f(a)$$

Lemma

$L(B_\sigma) = \Lambda^\epsilon(\sigma)$ for all σ.

A context-free grammar for $\Lambda(a)$

Consider the following context free grammar. Non terminals are B_σ for each $\emptyset \neq \sigma \subseteq H$.

$$B_\sigma \to \epsilon \qquad \text{for all } \sigma$$

$$B_\sigma \to B_{\sigma'} a B_{\sigma'} \qquad \text{for all } a \in \sigma \text{ and } \sigma' = \sigma \cup f(a)$$

Lemma

$L(B_\sigma) = \Lambda^\epsilon(\sigma)$ for all σ.

Example

$$B_p \to B_{\{p,r\}}\, p\, B_{\{p,r\}}$$

$$B_{\{p,r\}} \to B_{\{p,r\}}\, p\, B_{\{p,r\}}$$

$$B_{\{p,r\}} \to B_{\{r,p,s\}}\, r\, B_{\{r,p,s\}}$$

From the grammar to regular expressions

- Construct regular expressions for each $L(B_\sigma)$ starting with $\sigma = H$ then each σ of size $|H| - 1$ then $|H| - 2$ and so on.
- Each production rule from B_σ is either self-recursive or appeals to a $B_{\sigma'}$ with $|\sigma'| > |\sigma|$.
- Suppose we have constructed regular expressions r_μ for each $|\mu| > |\sigma|$. Assume $\sigma = \{a_1, a_2, b_1, b_2\}$ and the productions from B_σ are

$$B_\sigma \rightarrow B_\sigma a_1 B_\sigma \mid B_\sigma a_2 B_\sigma \mid B_\gamma b_1 B_\gamma \mid B_\eta b_2 B_\eta.$$

  Then $r_\sigma = (a_1 \mid a_2 \mid r_\gamma b_1 r_\gamma \mid r_\eta b_2 r_\eta)^*$.

How to cheat in favor of paper

$$B_p \rightarrow B_{\{p,r\}}\, p\, B_{\{p,r\}}$$
$$B_{\{p,r\}} \rightarrow B_{\{p,r\}}\, p\, B_{\{p,r\}}$$
$$B_{\{p,r\}} \rightarrow B_{\{r,p,s\}}\, r\, B_{\{r,p,s\}}$$

- $r_{\{r,p,s\}} = (r|p|s)^*$
- $r_{\{r,p\}} = ((r|p|s)^* r (r|p|s)^* \mid p)^*$
- $r_p = ((r|p|s)^* r (r|p|s)^* \mid p)^*\, p\, ((r|p|s)^* r (r|p|s)^* \mid p)^*$

Main result

Theorem

For any conservative groupoid H and any $h \in H$ the language $\Lambda(h)$ is regular.

Main result

Theorem

For any conservative groupoid H and any $h \in H$ the language $\Lambda(h)$ is regular.

Theorem

Any language $L \subseteq A^*$ recognized by a conservative groupoid lies in $\Sigma_2[<]$, i.e. it is a finite union of languages of the form

$$A_0^* a_1 A_1^* \dots A_{k-1}^* a_k A_k^*$$

with $A_i \subseteq A$ and $a_i \in A$.

Languages recognizable

Theorem

Suppose $L \subseteq A^*$ is recognizable by a conservative groupoid. Then

- $L \in \Sigma_2[<]$
- $L = L^+$
- *For all* $s, x, t \in A^*$ *it holds that* $sxt \in L \Rightarrow sx^2t \in L$.

Theorem

If $L \subseteq A^*$ *lies in* $\Sigma_1[<]$ *i.e. if it is a union of languages of the form*

$$A^* a_1 A^* \dots A^* a_k A^*$$

with $a_i \in A$ *then* L *is recognizable by a conservative groupoid.*

Open puzzles

Puzzle

Say that a conservative groupoid H can count up to t if there exists a word $u \in H^*$ s.t. $W(u^{t-1}) \neq W(u^t)$. For instance $\{r, p, s\}$ counts up to 2 since $W(rps) = \{r, s\}$ but $W(rpsrps) = \{r, p, s\}$.
We know that there is an H that counts up to t with $2t$ elements. This is not optimal since we also know a conservative groupoid that counts up to 6 with only 5 elements.

Open puzzles

Puzzle

Say that a conservative groupoid H can count up to t if there exists a word $u \in H^*$ s.t. $W(u^{t-1}) \neq W(u^t)$. For instance $\{r, p, s\}$ counts up to 2 since $W(rps) = \{r, s\}$ but $W(rpsrps) = \{r, p, s\}$.
We know that there is an H that counts up to t with $2t$ elements. This is not optimal since we also know a conservative groupoid that counts up to 6 with only 5 elements.

Puzzle

Find L and K such that L and K are recognizable by a conservative groupoid but $L \cap K$ is not.