CS 511, Fall 2024, Lecture Slides 07:

The Pigeonhole Principle in *Propositional Logic* and *First-Order Logic*

Assaf Kfoury

September 17, 2024

Reminder and Review

There are different versions of the **Pigeonhole Principle** (PHP), starting with the simplest and most common:

PHP_1: If $k+1$ or more objects are placed into k bins, where $k \geqslant 1$, then some bin contains two or more objects.

It is intuitively clear and easy to grasp, *but it still requires a proof!!*

Reminder and Review

There are different versions of the **Pigeonhole Principle** (PHP), starting with the simplest and most common:

PHP_1: If $k+1$ or more objects are placed into k bins, where $k \geqslant 1$, then some bin contains two or more objects.

It is intuitively clear and easy to grasp, *but it still requires a proof!!*

There are other versions of PHP, each requiring a slightly more complicated proof.

PHP_2: If m objects are placed into n bins, where $m > n \geqslant 1$, then some bin contains at least $\lceil \frac{m}{n} \rceil$ objects.

PHP_3: If m objects are placed into n bins, where $m > n \geqslant 1$, then some bin contains at most $\lfloor \frac{m}{n} \rfloor$ objects.

PHP_4: Let $B_1, B_2, \ldots, B_n$ be $n \geqslant 1$ bins, and $m_1, m_2, \ldots, m_n$ be n integers $\geqslant 0$. If $1 + \sum_{i=1}^{n} m_i$ objects are placed into these n bins, then some bin B_i contains at least $1 + m_i$ objects.

and there are still other more complicated versions of PHP.

Reminder and Review

The preceding versions of the **Pigeonhole Principle** (PHP) are all relative to finite numbers. There are also infinite versions of PHP:

PHP_5: If *infinitely* many objects are placed into *finitely* many bins, then there will exist at least one bin having *infinitely* many objects placed into it.

PHP_6: If *uncountably* many objects are placed into *countably* many bins, then there will exist at least one bin having *uncountably* many objects placed into it.

PHP_7: Let A and B be arbitrary sets, possibly *infinite*.
If there exists an injection $f : A \to B$ but there is no bijection $g : A \to B$,
then there does not exist an injection $h : B \to A$.

and there are still other versions of **infinite PHP**.

Reminder and Review

The preceding versions of the **Pigeonhole Principle** (PHP) are all relative to finite numbers. There are also infinite versions of PHP:

PHP_5: If *infinitely* many objects are placed into *finitely* many bins, then there will exist at least one bin having *infinitely* many objects placed into it.

PHP_6: If *uncountably* many objects are placed into *countably* many bins, then there will exist at least one bin having *uncountably* many objects placed into it.

PHP_7: Let A and B be arbitrary sets, possibly *infinite*.
If there exists an injection $f : A \to B$ but there is no bijection $g : A \to B$, then there does not exist an injection $h : B \to A$.

and there are still other versions of **infinite PHP**.

There is an enormous amount of material on PHP that you will find on the Web, with many alternative proofs for the same version of PHP.

A good place where to start is the Wikipedia page: Pigeonhole principle.

Some Applications (fun and surprising!)

Proposition: In Boston, there are at least two people with the same number of hairs on their heads!

In Brookline, this is probably ***not*** the case!

Some Applications (fun and surprising!)

Proposition: In Boston, there are at least two people with the same number of hairs on their heads!

In Brookline, this is probably ***not*** the case!

Proof: Apply PHP, using the following numbers:

Boston population $> 690,000$

Brookline population $< 60,000$

Biological fact: every human has $\leqslant 150,000$ hairs on the head. $\square$

Some Applications (fun and surprising!)

Proposition: In Boston, there are at least two people with the same number of hairs on their heads!

In Brookline, this is probably ***not*** the case!

Proof: Apply PHP, using the following numbers:

Boston population $> 690,000$

Brookline population $< 60,000$

Biological fact: every human has $\leqslant 150,000$ hairs on the head. $\square$

Proposition: It is ***not*** possible to cover an 8×8 chessboard with two diagonally opposite corners removed, with domino pieces of size 1×2.

Some Applications (fun and surprising!)

Proposition: In Boston, there are at least two people with the same number of hairs on their heads!

In Brookline, this is probably ***not*** the case!

Proof: Apply PHP, using the following numbers:

Boston population $> 690,000$

Brookline population $< 60,000$

Biological fact: every human has $\leqslant 150,000$ hairs on the head. $\square$

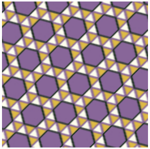
Proposition: It is ***not*** possible to cover an 8×8 chessboard with two diagonally opposite corners removed, with domino pieces of size 1×2.

Proof: There are 64 squares on the 8×8 chessboard, with 32 **white** and 32 **black**. With two diagonally opposite corners removed – say the two that are removed are **black** – there remains 32 **white** and 30 **black**. A domino piece of size 1×2 consists of 1 **white** and 1 **black**. Apply PHP: Think of each domino as being a "pigeon" and, there are 62 squares which we hope to cover with 31 non-overlapping dominos (*i.e.*, 31 pigeons), and think of each **black** square as being a "pigeonhole" and there are 30 **black** squares (*i.e.*, 30 pigeonholes). Hence, one **black** square must contain two (necessarily overlapping) dominos. $\square$

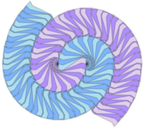
Some Applications (serious and still surprising!)

Theorem: Suppose that every point in the Cartesian plane is colored either ***red*** or ***blue***. Then for every real number $d > 0$, there are two points exactly at a distance d from each other that have the same color.

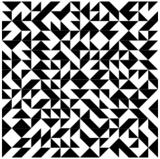
Different *2-colorings* of the plane (borrowed from Wikipedia Tessellation):



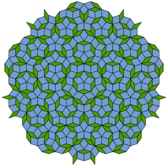
("brown" & "light brown")



("blue" & "purple")



("black" & "white")

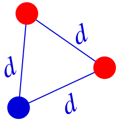


("blue" & "green")

Some Applications (serious and still surprising!)

Theorem: Suppose that every point in the Cartesian plane is colored either ***red*** or ***blue***. Then for every real number $d > 0$, there are two points exactly at a distance d from each other that have the same color.

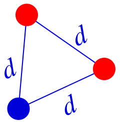
Proof: Consider an equilateral triangle whose side lengths are d. Place this triangle anywhere in the plane.



Some Applications (serious and still surprising!)

Theorem: Suppose that every point in the Cartesian plane is colored either ***red*** or ***blue***. Then for every real number $d > 0$, there are two points exactly at a distance d from each other that have the same color.

Proof: Consider an equilateral triangle whose side lengths are d. Place this triangle anywhere in the plane.



Because the triangle has three vertices and each point in the plane is only one of two different colors, by PHP at least two of the vertices must have the same color. These vertices are at distance d from each other, as required. $\square$

Some Applications (serious and still surprising!)

- **Hadwiger-Nelson Problem:** What is the minimum number k of colors required to color the plane such that any two points at distance 1 from each other have different colors.

  Call this minimum number k of colors the Hadwiger-Nelson (HN) number.

Some Applications (serious and still surprising!)

- **Hadwiger-Nelson Problem:** What is the minimum number k of colors required to color the plane such that any two points at distance 1 from each other have different colors.

  Call this minimum number k of colors the Hadwiger-Nelson (HN) number.

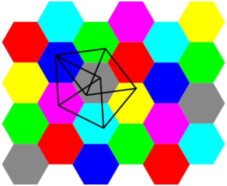
- By the theorem on slide 5, the HN number > 2.

Some Applications (serious and still surprising!)

- **Hadwiger-Nelson Problem:** What is the minimum number k of colors required to color the plane such that any two points at distance 1 from each other have different colors.

  Call this minimum number k of colors the Hadwiger-Nelson (HN) number.

- By the theorem on slide 5, the HN number > 2.
- There is a 7-*coloring* of the plane where any two points at distance 1 from each other have different colors (image from Wikipedia Hadwiger-Nelson Problem):

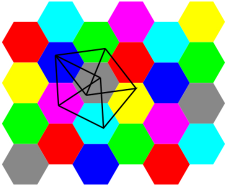


Some Applications (serious and still surprising!)

- **Hadwiger-Nelson Problem:** What is the minimum number k of colors required to color the plane such that any two points at distance 1 from each other have different colors.

  Call this minimum number k of colors the Hadwiger-Nelson (HN) number.

- By the theorem on slide 5, the HN number > 2.
- There is a 7-*coloring* of the plane where any two points at distance 1 from each other have different colors (image from Wikipedia Hadwiger-Nelson Problem):



- Hence, $2 <$ "HN number" $\leqslant 7$.
- **Open Problem**: The precise value of the HN number is still unknowm, but has been narrowed down to one of the numbers 5, 6 or 7.

The Pigeonhole Principle in Propositional Logic

We only consider the simplest version, as stated in slide 2 :

PHP_1: If n pigeons sit in $(n-1)$ holes, where $n \geqslant 2$, then some hole contains two or more pigeons.

The Pigeonhole Principle in Propositional Logic

We only consider the simplest version, as stated in slide 2:

PHP_1: If n pigeons sit in $(n-1)$ holes, where $n \geqslant 2$, then some hole contains two or more pigeons.

Let **PHP_1**(n) express **PHP_1** for the case of $n \geqslant 2$ pigeons. We can define a propositional wff φ_n, over the set of doubly-indexed variables:

$$\mathcal{Q} \stackrel{\text{def}}{=} \left\{ q_{i,j} \;\middle|\; i,j \in \{1,2,3,\ldots\} \right\}$$

whose satisfaction asserts the principle **PHP_1**(n):

$$\varphi_n \stackrel{\text{def}}{=} \underbrace{\bigwedge_{1 \leqslant i \leqslant n} \Big(\bigvee_{1 \leqslant j < n} q_{i,j} \Big)}_{(\spadesuit)} \rightarrow \underbrace{\bigvee_{1 \leqslant i < k \leqslant n} \Big(\bigvee_{1 \leqslant j < n} (q_{i,j} \wedge q_{k,j}) \Big)}_{(\diamondsuit)}$$

The Pigeonhole Principle in Propositional Logic

We only consider the simplest version, as stated in slide 2 :

PHP_1: If n pigeons sit in $(n-1)$ holes, where $n \geqslant 2$, then some hole contains two or more pigeons.

Let **PHP_1**(n) express **PHP_1** for the case of $n \geqslant 2$ pigeons. We can define a propositional wff φ_n, over the set of doubly-indexed variables:

$$\mathcal{Q} \stackrel{\text{def}}{=} \left\{ q_{i,j} \;\middle|\; i,j \in \{1,2,3,\ldots\} \right\}$$

whose satisfaction asserts the principle **PHP_1**(n):

$$\varphi_n \stackrel{\text{def}}{=} \underbrace{\bigwedge_{1 \leqslant i \leqslant n} \Big(\bigvee_{1 \leqslant j < n} q_{i,j} \Big)}_{(\spadesuit)} \rightarrow \underbrace{\bigvee_{1 \leqslant i < k \leqslant n} \Big(\bigvee_{1 \leqslant j < n} (q_{i,j} \wedge q_{k,j}) \Big)}_{(\diamondsuit)}$$

- satisfaction of $q_{i,j}$ = "pigeon i sits in hole j"
- satisfaction of $(\spadesuit)$ = "every pigeon sits in one of the holes"
- satisfaction of $(\diamondsuit)$ = "there are two pigeons sitting in the same hole"

NOTES for slide 19:

Once a student objected and proposed instead:

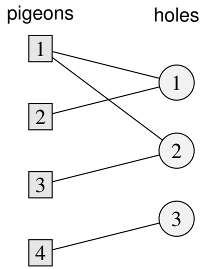
$$\varphi'_n \stackrel{\text{def}}{=} \underbrace{\Big(\bigwedge_{1 \leqslant i \leqslant n} \Big(\bigvee_{1 \leqslant j < n} q_{i,j} \Big)\Big)}_{(\spadesuit)} \wedge \underbrace{\Big(\bigwedge_{1 \leqslant i \leqslant n} \Big(\bigwedge_{1 \leqslant j < k < n} \neg (q_{i,j} \wedge q_{i,k}) \Big)\Big)}_{(\clubsuit)}$$

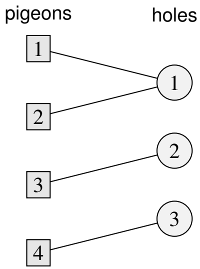
$$\to \underbrace{\bigvee_{1 \leqslant i < k \leqslant n} \Big(\bigvee_{1 \leqslant j < n} (q_{i,j} \wedge q_{k,j}) \Big)}_{(\diamondsuit)}$$

- satisfaction of $(\spadesuit)$ = "every pigeon sits in **at least** one of the holes"
- satisfaction of $(\clubsuit)$ = "every pigeon sits in **at most** one of the holes"
- satisfaction of $(\spadesuit) \wedge (\clubsuit)$ = "every pigeon sits in **exactly** one of the holes"
- satisfaction of $(\diamondsuit)$ = "there are two pigeons sitting in the same hole"

Question: Which of the two wff's, φ_n or φ'_n, formulates **PHP_1**(n)?

NOTES for slide 19:





On the left, graphical representation of truth assignment σ_1:

$$\sigma_1(q_{i,j}) \stackrel{\text{def}}{=} \begin{cases} \textit{true} & \text{if } (i,j) \in \{(1,1),(1,2),(2,1),(3,2),(4,3)\}, \\ \textit{false} & \text{otherwise.} \end{cases}$$

On the right, graphical representation of truth assignment σ_2:

$$\sigma_2(q_{i,j}) \stackrel{\text{def}}{=} \begin{cases} \textit{true} & \text{if } (i,j) \in \{(1,1),\cancel{(1,2)},(2,1),(3,2),(4,3)\}, \\ \textit{false} & \text{otherwise.} \end{cases}$$

Both $\sigma_1 \models (\spadesuit)$ and $\sigma_2 \models (\spadesuit)$, whereas $\sigma_1 \not\models (\clubsuit)$ and $\sigma_2 \models (\clubsuit)$.

NOTES for slide 19:

- In fact, φ_n is good enough to formally express **PHP_1**(n).
- You can take that φ'_n is more than needed to formally express **PHP_1**(n).

A little more precisely, let A be the set of n pigeons and B the set of $(n-1)$ holes, with $n \geqslant 2$.

- φ_n formalizes:
  - "A binary relation $R \subseteq A \times B$ cannot be injective", or also,
  - "A unary (possibly multi-valued) function $f : A \to B$ cannot be injective".
- φ'_n formalizes:
  - "A binary uni-valued relation $R \subseteq A \times B$ cannot be injective", or also,
  - "A unary uni-valued function $f : A \to B$ cannot be injective".

The Pigeonhole Principle in First-Order Logic

- We now want a first-order sentence ψ in the signature $\Sigma = \{R, c\}$ where R is a binary relation symbol and c is a constant symbol, such that:

  Satisfaction of ψ by a Σ-structure $\mathcal{M}_n \stackrel{\text{def}}{=} (\{1, 2, \ldots, n\}, R^{\mathcal{M}_n}, c^{\mathcal{M}_n})$ is equivalent to asserting **PHP_1**(n).

The Pigeonhole Principle in First-Order Logic

- We now want a first-order sentence ψ in the signature $\Sigma = \{R, c\}$ where R is a binary relation symbol and c is a constant symbol, such that:

  Satisfaction of ψ by a Σ-structure $\mathcal{M}_n \stackrel{\text{def}}{=} (\{1, 2, \ldots, n\}, R^{\mathcal{M}_n}, c^{\mathcal{M}_n})$ is equivalent to asserting **PHP_1**(n).

  Reminder: Given a binary predicate $P \subseteq A \times A$ on a set A:

  1. Dom$(P) \stackrel{\text{def}}{=} \{\, a \mid (a, b) \in P$ for some $b \in A \,\}$,
  2. Ran$(P) \stackrel{\text{def}}{=} \{\, b \mid (a, b) \in P$ for some $a \in A \,\}$,
  3. P is *many-one* iff there are $a_1, a_2 \in A$ such that $a_1 \neq a_2$ and $(a_1, b), (a_2, b) \in P$.

The Pigeonhole Principle in First-Order Logic

- We now want a first-order sentence ψ in the signature $\Sigma = \{R, c\}$ where R is a binary relation symbol and c is a constant symbol, such that:

  Satisfaction of ψ by a Σ-structure $\mathcal{M}_n \stackrel{\text{def}}{=} (\{1, 2, \ldots, n\}, R^{\mathcal{M}_n}, c^{\mathcal{M}_n})$ is equivalent to asserting **PHP_1**(n).

  Reminder: Given a binary predicate $P \subseteq A \times A$ on a set A:

  1. Dom(P) $\stackrel{\text{def}}{=} \{\, a \mid (a, b) \in P$ for some $b \in A \,\}$,
  2. Ran(P) $\stackrel{\text{def}}{=} \{\, b \mid (a, b) \in P$ for some $a \in A \,\}$,
  3. P is *many-one* iff there are $a_1, a_2 \in A$ such that $a_1 \neq a_2$ and $(a_1, b), (a_2, b) \in P$.

- Here is a possible first-order formulation of ψ:

$$\psi \stackrel{\text{def}}{=} \big(\forall x\, \neg R(x, c)\big) \;\wedge\; \big(\forall x\, \exists y\, R(x, y)\big) \rightarrow \exists v\, \exists w\, \exists y \big(\neg(v \approx w) \wedge R(v, y) \wedge R(w, y)\big)$$

In words: *"If there is an element c not in the range of total relation R, then R is many-one"*

The Pigeonhole Principle in First-Order Logic

- We now want a first-order sentence ψ in the signature $\Sigma = \{R, c\}$ where R is a binary relation symbol and c is a constant symbol, such that:

  Satisfaction of ψ by a Σ-structure $\mathcal{M}_n \overset{\text{def}}{=} (\{1, 2, \ldots, n\}, R^{\mathcal{M}_n}, c^{\mathcal{M}_n})$ is equivalent to asserting **PHP_1**(n).

  Reminder: Given a binary predicate $P \subseteq A \times A$ on a set A:

  1. Dom$(P) \overset{\text{def}}{=} \{\, a \mid (a, b) \in P$ for some $b \in A \,\}$,
  2. Ran$(P) \overset{\text{def}}{=} \{\, b \mid (a, b) \in P$ for some $a \in A \,\}$,
  3. P is *many-one* iff there are $a_1, a_2 \in A$ such that $a_1 \neq a_2$ and $(a_1, b), (a_2, b) \in P$.

- Here is a possible first-order formulation of ψ:

$$\psi \overset{\text{def}}{=} \big(\forall x \,\neg R(x, c)\big) \;\wedge\; \big(\forall x \,\exists y \, R(x, y)\big) \to \exists v \,\exists w \,\exists y \big(\neg(v \approx w) \wedge R(v, y) \wedge R(w, y)\big)$$

  In words: *"If there is an element c not in the range of total relation R, then R is many-one"*

  Remark: If $\big(\forall x \,\neg R(x, c)\big)$ is omitted to obtain a new sentence θ, there is a structure $\mathcal{M}_n$ satisfying $\big(\forall x \,\exists y \, R(x, y)\big)$ but not $\exists v \,\exists w \,\exists y \big(\neg(v \approx w) \wedge R(v, y) \wedge R(w, y)\big)$, in which case $\mathcal{M}_n \not\models \theta$ and **PHP_1**(n) is not enforced by satisfaction of θ.

- Advantage of a first-order formulation over a propositional formulation : one FOL wff ψ instead of infinitely many PL wff's $\{\varphi_2, \varphi_3, \ldots\}$

The Pigeonhole Principle: PL vs. FOL

- **Exercise**:
  1. Translate ψ into a PL wff ψ_n which depends on an additional parameter $n \geqslant 2$. (ψ represents an infinite family of PL wff's, one ψ_n for every $n \geqslant 2$.)
     Hint: Consider replacing every "$\forall$" by a "$\bigwedge$" and every "$\exists$" by a "$\bigvee$".
  2. Compare φ_n and ψ_n.
     Hint: They are very close to each other.

- **Exercise**:
  1. Use an automated proof-assistant (*e.g.*, Isabelle, Coq, etc.) to establish that the FOL wff ψ is valid.
  2. Use a SAT solver to establish that the PL wff's φ_2, φ_3, and φ_4 are each valid.
  3. Compare the performances in part 1 and part 2.

The Pigeonhole Principle: PL vs. FOL

- **Fact**:
  A resolution proof (*Lecture Slides 11*) of φ_n or ψ_n is possible but does not help:
  any resolution proof of φ_n or ψ_n has size at least $\Omega(2^n)$ (bad news!).
- **Fact**:
  There are proofs of φ_n and ψ_n using what is called extended resolution
  (not covered this semester) which have size $\mathcal{O}\left(n^4\right)$.
- **Fact**:
  There are Hilbert-style proofs of φ_n and ψ_n which have size at most $\mathcal{O}\left(n^{20}\right)$
  (not really good news!).

The Pigeonhole Principle: PL vs. FOL

- **Fact**:
  A resolution proof (*Lecture Slides 11*) of φ_n or ψ_n is possible but does not help:
  any resolution proof of φ_n or ψ_n has size at least $\Omega(2^n)$ (bad news!).
- **Fact**:
  There are proofs of φ_n and ψ_n using what is called extended resolution
  (not covered this semester) which have size $\mathcal{O}\left(n^4\right)$.
- **Fact**:
  There are Hilbert-style proofs of φ_n and ψ_n which have size at most $\mathcal{O}\left(n^{20}\right)$
  (not really good news!).

- **Exercise**:
  Use the method of analytic tableaux (*Lecture Slides 10*) to establish the validity
  of the PL wff's φ_n and ψ_n. Give an estimate of the complexity.

The Pigeonhole Principle in FOL – once more

- We define another first-order sentence ψ' in the signature $\Sigma' = \{f, c\}$ where f is a unary function symbol and c is a constant symbol, such that:

  *Every Σ'-structure $\mathcal{N}_n \stackrel{\text{def}}{=} (\{1, 2, \ldots, n\}, f^{\mathcal{N}_n}, c^{\mathcal{N}_n})$ is a model of ψ' and the interpretation of ψ' in $\mathcal{N}_n$ expresses **PHP_1**(n).*

The Pigeonhole Principle in FOL – once more

- We define another first-order sentence ψ' in the signature $\Sigma' = \{f, c\}$ where f is a unary function symbol and c is a constant symbol, such that:

  *Every Σ'-structure $\mathcal{N}_n \stackrel{\text{def}}{=} (\{1, 2, \ldots, n\}, f^{\mathcal{N}_n}, c^{\mathcal{N}_n})$ is a model of ψ' and the interpretation of ψ' in $\mathcal{N}_n$ expresses **PHP_1**(n).*

- Here is a possible first-order formulation of ψ':

$$\boxed{\psi' \stackrel{\text{def}}{=} \big(\forall x.\ \neg\big(f(x) \approx c\big)\big) \to \exists v\, \exists w\, \big(\neg(v \approx w) \land f(v) \approx f(w)\big)}$$

(THIS PAGE INTENTIONALLY LEFT BLANK)