CS 40
FOUNDATIONS OF CS

Summer 2024

Session A

Thursday's learning goals

- Application of propositional logic to logic puzzles
- Define data types: set, n-tuple, string (over specific alphabet)
- Define sets and functions in multiple ways
- Define a predicate over a finite domain using a table of values and as properties
- Determine the truth value of the proposition resulting from evaluating a predicate
- Describe the set of domain elements that make a predicate with one input evaluate to true.
- Practice translations from predicate logic to English
- Practice combinations of $\wedge$, $\rightarrow$ in conjunction with universal and existential quantifiers
- State and apply DeMorgan's law for quantified statements.

Logic Puzzles

Knaves: Always Lie

An island has two types of inhabitants: Knights and Knaves. You meet two people on the island: **A** and **B**.

A says: “ I am a knave or B is a knight”
B says nothing

What are A and B?

Knights: Always tell the truth

Logic Puzzles

Define propositions:

✓ p : **A is a knight**

✓ q : **B is a knight**

Knaves: Always Lie

Knights: Always tell the truth

A says: " I am a knave or B is a knight"

B says nothing $\sim p \vee q$

What are A and B?

The puzzle can be completely described by which of the following statements?

A. $p \to \neg q$

B. $p \to (\neg p \vee q)$

C. $p \leftrightarrow (\neg p \vee q)$ ✓

D. More than one of the above

E. None of the above

~~Recap:~~

$$(p \longrightarrow \sim p \vee q) \wedge \underbrace{(\sim p \longrightarrow \sim(\sim p \vee q))}_{\text{contrastve}}$$

$$(p \rightarrow \sim p \vee q) \wedge (\sim p \vee q \longrightarrow p)$$

$$p \Longleftrightarrow \sim p \vee q$$

Truth Table

p	q	$\sim p$	$\sim p \vee q$	$p \Leftrightarrow \sim p \vee q$
T	T	F	T	T
T	F	F	F	F
F	T	T	T	F
F	F	T	T	F

$\Rightarrow$ True only if

p: A is knight

q: B is knight

Logic Puzzles

Define propositions:
p : A is a knight
q : B is a knight

Knaves: Always Lie

Knights: Always tell the truth

A says: " I am a knave or B is a knight"
B says nothing
What are A and B?

What truth values for p and q make the proposition $p \leftrightarrow (\neg p \lor q)$ true?

A. TT
B. TF
C. FT
D. FF

$S_1 =$ Types: set

$S_1 = \{1, 1, 2, 3\}$ $S_2 = \{1, 2, 3\}$ are S_1 & S_2 equal?

1, 2, 3

$S = \{1, 2, 3\}$

empty set ϕ

$\{\ \}$

$\{\ \}$ → Roster Representation

→ Set containing an empty set

↓

Roster notation for this set?

$= \{\{\}\}$

$1 \in S$

$2 \in S, 5 \notin S$

Term	Examples: (add additional examples from class)
set unordered collection of elements *Equal means agree on membership of all elements*	$7 \in \{43, 7, 9\}$ $2 \notin \{43, 7, 9\}$

Defining sets

$S = \{1, 2, 3, \ldots, 10\}$

$S = \{x \in \mathbb{Z} : 1 \le x \le 10\} \equiv \{x : x \in \mathbb{Z},\ 1 \le x \le 10\}$

or |

integers

- Roster method
- Set builder notation

Answer the questions below about the given examples:

$\{-1, 1\}$ $\quad \{0, 0\}$ $\quad \{-1, 0, 1\}$ $\quad \mathbb{Z}$ $\quad \mathbb{N} = \{x \in \mathbb{Z} \mid x \ge 0\}$ $\quad \emptyset$ $\quad \mathbb{Z}^+ = \{x \in \mathbb{Z} \mid x > 0\}$

1 2 3 4 5 6 7

empty set

set of all natural nos

+ve integers

- *Which of the sets above are defined using the roster method?* 1, 2, 3
- *Which sets are defined using set builder notation?* 5, 7
- *Which of the sets above have 0 as an element?* 2, 3, 5 ✓4 ?
- *Can you write any of the sets above more simply?* $\{0, 0\} = \{0\}$

Defining functions

A function is defined by

(1) domain Nonempty set

(2) codomain or target Nonempty set

(3) rule assigning each element in the domain exactly one element in the codomain Table, formula, etc.

Notation:

domain $\xrightarrow{\text{input}}$ $\boxed{f}$ $\xrightarrow{\text{output}}$ codomain

Rule : $f : \mathbb{N} \longrightarrow \mathbb{Z}$

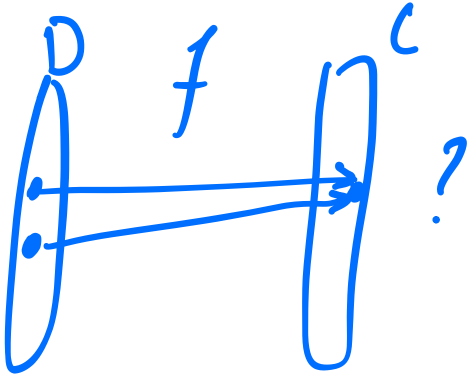
$f(x) = 3x + 1$

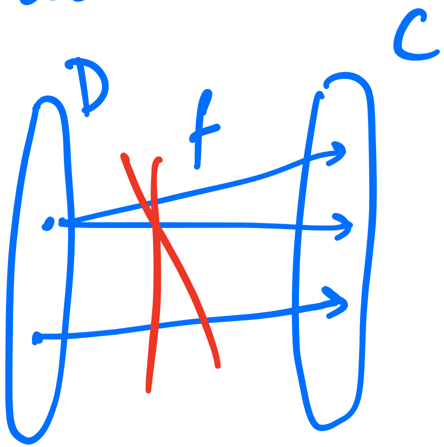
Functions must be well-defined

For a function to be **well-defined** it must be the case that: domain, codomain are defined

each element maps to exactly one element.

/ the fn map each element in the domain to exactly one element in the codomain





Functions must be well-defined

Is the following function well-defined?

$$f(x) = \pm\sqrt{1 - x^2}$$

counter ex: take $x = 0$ $f(0) = 1$ & $f(0) = -1$

A. Yes
B. No, because the domain and codomain are not defined
C. No, because there is an error in the notation
D. No, because the function gives the same output for two different inputs
E. No, some other reason.

Predicates as functions

A **predicate** is a function from a given set (domain) to {T,F}
It can be specified by its input-output definition table **or by specifying the rule**
It can be applied, or **evaluated at**, an element of the domain.

Input	Output		
	$P(x)$	$N(x)$	$Mystery(x)$
x	$x > 0$	$x < 0$	
0	F	F	T
1	T	F	T
2	T	F	T
3	T	F	F
−4	F	T	F
−3	F	T	T
−2	F	T	F
−1	F	T	T

What is the domain of P(x)? $D = \{-4, -3, \ldots, 3\}$

$D = \{x \in \mathbb{Z} \mid -4 \le x \le 3\}$

$P : D \longrightarrow \{T, F\}$

$P(x) = T$ if $x > 0$ and F otherwise

Predicates as functions

Input x	$Mystery(x)$
0	T
1	T
2	T
3	F
-4	F
-3	T
-2	F
-1	T

Which of the following is a description of the rule that corresponds to the input-output definition table for *Mystery(x)* ?

A: “x is a non-zero number”
B: “x is a non-negative number”
C: “x is a number that is even and positive”
D: “x is a number that is not positive and is not negative”
E: None of the above

Predicates as functions

A **predicate** is a function from a given set (domain) to {T,F}
It can be specified by its input-output definition table or by specifying the rule
or by specifying its truth set: the elements of the domain at which the predicate evaluates to T

Input	Output		
	$P(x)$	$N(x)$	$Mystery(x)$
x	$x > 0$	$x < 0$	
0	F		T
1	T		T
2	T		T
3	T		F
-4	F		F
-3	F		T
-2	F		F
-1	F		T

Truth set for P(x) is $\{1, 2, 3\}$

Truth set for N(x) is $\{-4, \ldots, -1\}$

Truth set for *Mystery(x)* is $\{0, 1, 2, -3, -1\}$

Why is specifying the truth set of a predicate enough to define its rule?

$f : \mathbb{Z} \longrightarrow \{T, F, N\}$

Truth Set $= \{x \in \mathbb{Z} : x \geq 0\}$ Just this is not enough to define f.

but if co-domain $\{T, F\}$ $\longrightarrow$ then enough

Quantified statements

zyBook 2.4, Rosen p. 40-45

We can make claims about a set by saying which or how many of its elements satisfy a property. These claims are called quantified statements and use predicates.

	$P(x)$
1	T
2	T
3	T
4	T

✓

for all x $P(x)$

The **universal quantification** of $P(x)$ is the statement "$P(x)$ for all values of x in the domain" and is written $\forall x P(x)$.

The **existential quantification** of $P(x)$ is the statement "There exists an element x in the domain such that $P(x)$" and is written $\exists x P(x)$.

there exist x $P(x)$

Rewrite "for all x, P(x)" in its expanded form using logical connectives

$\mathbb{Z}^+$

$\forall x\, P(x) = P(1) \wedge P(2) \wedge P(3) \ldots$

Input	Output		
	$P(x)$	$N(x)$	$Mystery(x)$
x	$x > 0$	$x < 0$	
0	F		T
1	T		T
2	T		T
3	T		F
-4	F		F
-3	F		T
-2	F		F
-1	F		T

Rewrite "there exists x, P(x)" in its expanded form using logical connectives

Domain $\mathbb{Z}^+$

$\exists x\, P(x) = P(1) \vee P(2) \vee P(3) \ldots$

Translate to English

quantified statements over finite domains involving connectives

$\forall x(\ P(x) \lor N(x)\ )$ → is this a True statement

≡ For every value of x either P(x) is T or N(x) is T

Input	Output		
	$P(x)$	$N(x)$	$Mystery(x)$
x	$x > 0$	$x < 0$	
0	F	F	T
1	T	F	T
2	T	F	T
3	T	F	F
−4	F	T	F
−3	F	T	T
−2	F	T	F
−1	F	T	T

counter example

$\exists x(\ P(x) \rightarrow N(x)\ )$

≡ There exists a value x such that if P(x) then N(x)

is this a True statement?

'tness: x=0 P(x) F N(x) - P(x) → N(x) T

Strategy to prove :

$\forall x \; R(x)$

$\exists x \; R(x)$

• give a witness

Strategy to disprove

$\forall x \; R(x)$

• give a counter example

$\exists x \; R(x)$

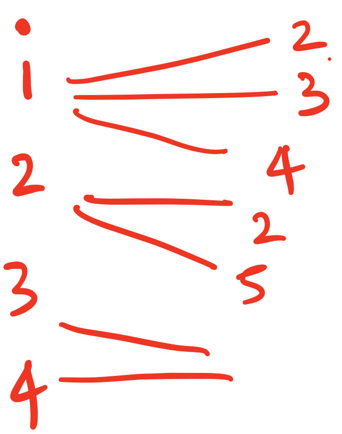
	$P(x)$	$N(x)$	$P(x) \to N(x)$
✓ 1	T	F	F
✓ 2	T	F	F
3	T	F	F
4	T	F	F

disprove $\exists x \; P(x) \to N(x)$

enumerate all possibilities

↳ show not true for everything

$\forall x \; \exists y \in f(x) \; P(y)$ → domain 2, 3, 4, 5



y	$P(y)$
2	T
3	F
4	F
2	T
5	F

→ explain this with English examples

Thursday's learning goals

- Practice translations
- Counterexample and witness-based arguments for predicates with finite domains
- Evaluate universal and existential statements about finite domains
- Define sets and functions recursively
- Practice with properties of recursively defined sets and functions in a new application domain: Bioinformatics!
- Use predicates with set of tuples as their domain to relate values to one another

Translate to Symbolic Form

- S(x): "x is a student in this class"
- M(x): "x has visited Mexico"

Domain: All the people in the world

Translate the statement to symbolic form

"Some student in this class has visited Mexico"

use predicates $S(x), M(x)$

use $\forall$ $\exists$

$\exists x \; S(x) \wedge M(x)$

Translate to Symbolic Form

- S(x): “x is a student in this class”
- M(x): “x has visited Mexico”

Domain: All the people in the world

Translate the statement to symbolic form

“Every student in this class has visited Mexico”

$\forall x \; S(x) \wedge M(x)$ ✗ → every person in the world is a student in this class & has visited Mexico

✓ $\forall x \; S(x) \rightarrow M(x)$

Quantifier De Morgan's Laws

zybook 2.6

Rosen p. 45

De Morgan's : $\sim(p \vee q) = \sim p \wedge \sim q$ $\quad \sim(p \wedge q) = \sim p \vee \sim q$

Statements involving predicates and quantifiers are **logically equivalent** means they have the same truth value no matter which predicates (domains and functions) are substituted in.

Quantifier version of De Morgan's laws: $\boxed{\neg \forall x P(x) \equiv \exists x (\neg P(x))}$ $\quad \boxed{\neg \exists x Q(x) \equiv \forall x (\neg Q(x))}$

ex in English

Example: $\forall x\ p(x) \vee N(x)$ is a false universal quantification.

The negation of the statement is: $\sim \forall x \left(p(x) \vee N(x)\right)$.

The negated statement is logically equivalent to $\exists x\ \sim p(x) \wedge \sim N(x)$

$\exists x \sim (p(x) \vee N(x))$

$\rightarrow \exists \sim p(x) \wedge \sim N(x)$

$\sim \forall x \; s(x)$ → not (all people in the world are a student in this class)

$= \exists x \; \sim s(x)$

→ $\exists$ a person in the world who is not a student in this class.

$\sim \exists x \; S(x)$ → people in the world who have landed on the sun

$\equiv \forall x \; \sim S(x)$

Apply DeMorgan's Law

In the following question, the domain is a set of patients in a clinical study. Define the following predicates:

P(x): x was given the placebo
D(x): x was given the medication
M(x): x had migraines

Translate each statement into a logical expression. Then negate the expression by adding a negation operation to the beginning of the expression. Apply De Morgan's law until each negation operation applies directly to a predicate and then translate the logical expression back into English.

Every patient who took the placebo had migraines.

$\forall x \; P(x) \longrightarrow M(x)$

Negate: $\sim \forall x \; P(x) \longrightarrow M(x)$

$\Rightarrow \exists x \sim (P(x) \longrightarrow M(x)) \Rightarrow \exists x \; P(x) \wedge \sim M(x)$?

$\Rightarrow \exists x \sim (\sim P(x) \vee M(x)) \Longrightarrow$ DeMorgan's

Counterexamples and Witnesses

Definition: An element for which $P(x)$ is F is called a **counterexample** of $\forall x P(x)$.

Definition: An element for which $P(x)$ is T is called a **witness** of $\exists x P(x)$.

Input	Output		
	$P(x)$	$N(x)$	$Mystery(x)$
x	$x > 0$	$x < 0$	
0	F		T
1	T		T
2	T		T
3	T		F
-4	F		F
-3	F		T
-2	F		F
-1	F		T

For a predicate E(x), which of the following is a valid proof strategy:

A: We can prove that $\forall x E(x)$ using a witness. is false

B: We can prove that $\forall x E(x)$ using a counterexample.

C: We can prove that $\exists x E(x)$ using a witness. is false

D: We can prove that $\exists x E(x)$ using a counterexample

E: More than one of the above

The **universal quantification** of $P(x)$ is the statement "$P(x)$ for all values of x in the domain" and is written $\forall x P(x)$. An element for which $P(x) = F$ is called a **counterexample** of $\forall x P(x)$.

Which of the following is a true statement?

A. $\forall x \; P(x)$ ~~T~~ F ✓

B. $\forall x \; N(x)$ F

C. $\forall x \; Mystery(x)$ F

D. All of the above

E. None of the above ✓

Input	Output		
	$P(x)$	$N(x)$	$Mystery(x)$
x	$x > 0$	$x < 0$	
0	F		T
1	T		T
2	T		T
3	T		F
-4	F		F
-3	F		T
-2	F		F
-1	F		T

The **existential quantification** of $P(x)$ is the statement "There exists an element x in the domain such that $P(x)$" and is written $\exists x P(x)$. An element for which $P(x) = T$ is called a **witness** of $\exists x P(x)$.

Which of the following is a true statement?

A. ∃xP(x) T

B. ∃xN(x)

C. ∃xMystery(x)

D. All of the above

E. None of the above

Input	Output		
	$P(x)$	$N(x)$	$Mystery(x)$
x	$x > 0$	$x < 0$	
0	F		T
1	T		T
2	T		T
3	T		F
-4	F	T	F
-3	F		T
-2	F		F
-1	F		T

Counterexamples and Witnesses

for quantified statements over finite domains involving connectives

Which of these is true?

A: $\forall x(\ P(x) \vee N(x)\ )$ F

B: $\exists x(\ P(x) \rightarrow N(x)\ )$ T

C: $\forall x(\ P(x) \oplus Mystery(x)\ )$ F

D: More than one of the above

E: None of the above

$P(x) \oplus Mystery(x)$: T, F

$P(x) \rightarrow N(x)$: T

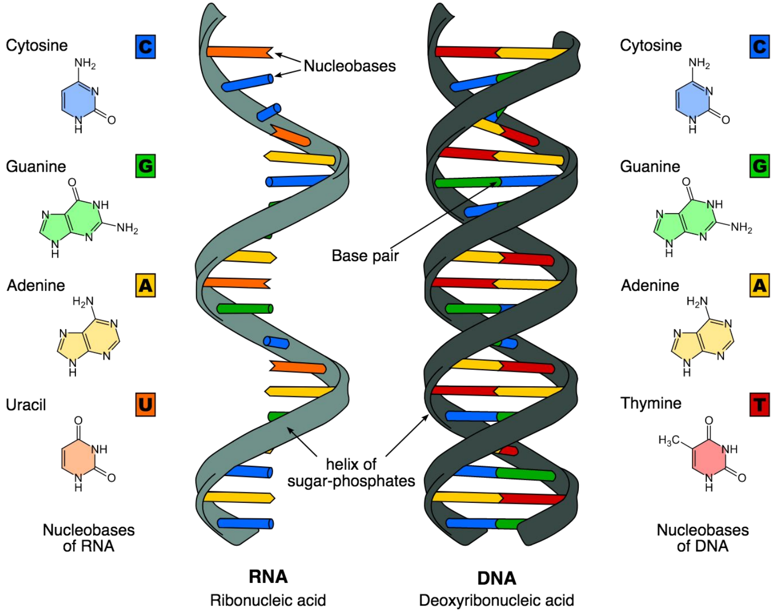
$P(x) \vee N(x)$: F, T, T, T, T, T, T, T

Input	Output		
x	$P(x)$ $x > 0$	$N(x)$ $x < 0$	$Mystery(x)$
0	F	F	T
1	T	F	T
2	T	F	T
3	T	F	F
-4	F	T	F
-3	F	T	T
-2	F	T	F
-1	F	T	T

Next Class

RNA strands as strings

To be Continued



Each RNA strand is a **string** whose symbols are elements of the set $B = \{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{U}\}$.

Definition by recursion

New! Recursive Definitions of Sets: The set S (pick a name) is defined by:

Basis Step:	Specify finitely many elements of S
Recursive Step:	Give a rule for creating a new element of S from known values existing in S, and potentially other values.

Definition The set of RNA strands S is defined (recursively) by:

Basis Step:	$\mathtt{A} \in S, \mathtt{C} \in S, \mathtt{U} \in S, \mathtt{G} \in S$
Recursive Step:	If $s \in S$ and $b \in B$, then $sb \in S$

Two different RNA strands:

Defining functions recursively when domain is recursively defined

Definition (Of a function, recursively) A function *rnalen* that computes the length of RNA strands in S is defined by:

$$\begin{array}{lll} & & rnalen : S \to \mathbb{Z}^+ \\ \text{Basis Step:} & \text{If } b \in B \text{ then} & rnalen(b) = 1 \\ \text{Recursive Step:} & \text{If } s \in S \text{ and } b \in B\text{, then} & rnalen(sb) = 1 + rnalen(s) \end{array}$$

The domain of *rnalen* is ____________________. The codomain of *rnalen* is ____________________.

$rnalen(\texttt{ACU}) =$ ____

Recall: Each RNA strand is a string whose symbols are elements of the set $B = \{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{U}\}$. The **set of all RNA strands** is called S. The function *rnalen* that computes the length of RNA strands in S is:

$$
\begin{array}{llll}
 & & rnalen : S & \to \mathbb{Z}^+ \\
\text{Basis Step:} & \text{If } b \in B \text{ then} & rnalen(b) & = 1 \\
\text{Recursive Step:} & \text{If } s \in S \text{ and } b \in B, \text{ then} & rnalen(sb) & = 1 + rnalen(s)
\end{array}
$$

Example predicates on S

$H(s) = T$	Truth set of H is ____________
$L_3(s) = \begin{cases} T & \text{if } rnalen(s) = 3 \\ F & \text{otherwise} \end{cases}$	Strand where L_3 evaluates to T is e.g.________ Strand where L_3 evaluates to F is e.g. ________
$F_{\mathtt{A}}$ is defined recursively by: Basis step: $F_{\mathtt{A}}(\mathtt{A}) = T$, $F_{\mathtt{A}}(\mathtt{C}) = F_{\mathtt{A}}(\mathtt{G}) = F_{\mathtt{A}}(\mathtt{U}) = F$ Recursive step: If $s \in S$ and $b \in B$, then $F_{\mathtt{A}}(sb) = F_{\mathtt{A}}(s)$	Strand where $F_{\mathtt{A}}$ evaluates to T is e.g.________ Strand where $F_{\mathtt{A}}$ evaluates to F is e.g. ________
$P_{\mathtt{AUC}}$ is defined as the predicate whose truth set is the collection of RNA strands where the string AUC is a substring (appears inside s, in order and consecutively)	Strand where $P_{\mathtt{AUC}}$ evaluates to T is e.g.________ Strand where $P_{\mathtt{AUC}}$ evaluates to F is e.g. ________

$H(s) = T$

$$L_3(s) = \begin{cases} T & \text{if } rnalen(s) = 3 \\ F & \text{otherwise} \end{cases}$$

$F_{\mathtt{A}}$ is defined recursively by:

Basis step: $F_{\mathtt{A}}(\mathtt{A}) = T$, $F_{\mathtt{A}}(\mathtt{C}) = F_{\mathtt{A}}(\mathtt{G}) = F_{\mathtt{A}}(\mathtt{U}) = F$

Recursive step: If $s \in S$ and $b \in B$, then $F_{\mathtt{A}}(sb) = F_{\mathtt{A}}(s)$

$P_{\mathtt{AUC}}$ is defined as the predicate whose truth set is the collection of RNA strands where the string AUC is a substring (appears inside s, in order and consecutively)

A true universal quantification is:

A false universal quantification is:

A true existential quantification is:

A false existential quantification is:

Predicates with multiple inputs

Defining Sets

- Roster method, set builder, recursive
- **New** Applying operations to other sets

Definition (Rosen p. 123) Let A and B be sets. The **Cartesian product** of A and B, denoted $A \times B$, is the set of all ordered pairs (a, b) where $a \in A$ and $b \in B$

$$A \times B = \{(a, b) \mid a \in A \text{ and } b \in B\}$$

$B = \{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{U}\}$

Set	Example elements in this set:
Fill in possible set	$(\mathtt{A}, \mathtt{C})$ $(\mathtt{U}, \mathtt{U})$
$B \times \{-1, 0, 1\}$	Fill in example elements
$\{-1, 0, 1\} \times B$	Fill in example elements
Fill in possible set	$(0, 0, 0)$
$\{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{U}\} \circ \{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{U}\}$	Fill in example elements
Fill in possible set	$\mathtt{GGGG}$

Cartesian Products and Predicates

Notation: for a predicate P with domain $X_1 \times \cdots \times X_n$ and a n-tuple $(x_1, \ldots, x_n)$ with each $x_i \in X$, we write $P(x_1, \ldots, x_n)$ to mean $P(\ (x_1, \ldots, x_n)\ )$.

L with domain $S \times \mathbb{Z}^+$ is defined by, for $s \in S$ and $n \in \mathbb{Z}^+$,

$$L(s,n) = \begin{cases} T & \text{if } rnalen(s) = n \\ F & \text{otherwise} \end{cases}$$

Element where L evaluates to T: ________

Element where L evaluates to F: ________

Cartesian Products and Predicates

BC with domain $S \times B \times \mathbb{N}$ is defined by, for $s \in S$ and $b \in B$ and $n \in \mathbb{N}$,

$$BC(s,b,n) = \begin{cases} T & \text{if } basecount(s,b) = n \\ F & \text{otherwise} \end{cases}$$

Which of these is a witness that proves that $\exists t\ BC(t)$ is true?

A. `G`
B. (`GA`, 2)
C. (`GG`, `C`, 0)
D. None of the above, but something else works.
E. None of the above, because the statement is false.

The **existential quantification** of $P(x)$ is the statement "There exists an element x in the domain such that $P(x)$" and is written $\exists x P(x)$. An element for which $P(x) = T$ is called a **witness** of $\exists x P(x)$.

Cartesian Products and Predicates

BC with domain $S \times B \times \mathbb{N}$ is defined by, for $s \in S$ and $b \in B$ and $n \in \mathbb{N}$,

$$BC(s,b,n) = \begin{cases} T & \text{if } basecount(s,b) = n \\ F & \text{otherwise} \end{cases}$$

Which of these is a counterexample that proves that $\forall (s,b,n) \; (\; BC(s,b,n) \;)$ is false?

A. (`G`, `A` , 1)
B. (`GC`, `A` , 3)
C. (`GG`, `G` , `2`)
D. None of the above, but something else works.
E. None of the above, because the statement is true.

The **universal quantification** of $P(x)$ is the statement "$P(x)$ for all values of x in the domain" and is written $\forall x P(x)$. An element for which $P(x) = F$ is called a **counterexample** of $\forall x P(x)$.

Predicate	Domain	Example domain element where predicate is T
$basecount(s, b) = 3$		
$basecount(s, \mathtt{A}) = n$		
$\exists n \in \mathbb{N}\ (basecount(s, b) = n)$		
$\forall b \in B\ (basecount(s, b) = 1)$		

Summary

To prove that the **universal quantification**

$$\forall x P(x)$$

is **true** when the predicate P has a finite domain, evaluate P(x) at each domain element to confirm it is T.

To prove that the **universal quantification**

$$\forall x P(x)$$

is **false**, we find a counterexample: an element in the domain for which P(x) is false.

To prove that the **existential quantification**

$$\exists x P(x)$$

is **true**, we find a witness: an element in the domain for which P(x) is true.

To prove that the **existential quantification**

$$\exists x P(x)$$

is **false** when the predicate P has a finite domain, evaluate P(x) at each domain element to confirm it is F.