Lesson 1: Set Builder Notation & Tuples

Computer Science II
Mrs. Staffen

1. Representing Sets: SET-BUILDER Notation

1.1 Motivation

1.2 Syntax of set builder notation

1.3 Meaning of set builder notation

1.4 Example and Practice

1. Representing Sets: SET-BUILDER Notation

1.1 SET builder notation (MOTIVATION)

- Sometimes we don't want to enumerate a set. Instead we want to describe the sets using *properties* or *conditions*.

- We may want to refer to the set of big cities of the USA. Note *big* is a property of a city. Assume we have a function *big(x)* that maps a city *x* to a truth value (i.e., true or false). Also let *citiesOfUSA* be the set of cities of USA. Then the set of big cities can be described using the form:

$$\{\, x : big(x)=\text{true}\ \textbf{and}\ x \in citiesOfUSA \,\}$$

1. Representing Sets: SET-BUILDER Notation

1.1 SET builder notation (MOTIVATION)

$$\{x\text{: } big(x) = \text{true } \textbf{and } x \in citiesOfUSA\}$$

- The part in red font is called a *property* or condition. It is a proper combination of expressions using **logical connectives** such as **and**.
- It is **read** as the set consisting of every *x* (a variable referring to anything) such that *big(x)* is true and *x* is a member of *citiesOfUSA*.
- This form of representing a set is called *set builder notation*.

1. Representing Sets: SET-BUILDER Notation

1.2 Syntax of set builder notation

Set builder notation is of the form

$$\{\, x : \textit{condition} \}$$

- x is any name. It is a *variable* referring to an element of the set, and
- *condition* is an expression whose value is true or false or a number of such expressions combined by logical connectives (*and*, *or*, *if*, *negation*). Normally, x is a *variable* in *condition*. The *condition* tells us what the (property of) x is.
- Note the fixed symbols and their positions: **{   :   }**

It is **read** as the set consisting of every x that
makes the *condition* true (or satisfies the *condition*).

1.2 Syntax of set builder notation - example

In the set builder notation form

$$\{\, x : condition \}$$

- x is any name (without a space in it)
- *condition* is an expression whose value is true or false or a number of such expressions combined by logical connectives (and, or, if).

Consider example $\{city: big(city)=\text{true } \textbf{and } city \in citiesOfUSA\}$

- *city* is a name,
- $big(city)=\text{true}$ is an expression with truth value,
- $city \in citiesOfUSA$ is an expression with truth value.
- Logical connective "**and**" combines them: $big(city)=\text{true } \textbf{and } city \in citiesOfUSA$

1. Representing Sets: SET-BUILDER Notation

1.1 Motivation

1.2 Syntax of set builder notation

1.3 Meaning of set builder notation

1.4 Example and Practice

1.3 Meaning of set builder notation

$$\{ \, x : condition \}$$

It is **read** as the set consisting of every *x* that makes the *condition* true.

Consider example $\{city\text{: } big(city)\text{=true } \textbf{and } city \in citiesOfUSA\}$

It is **read** as

1.3 Meaning of set builder notation

$$\{\, x : \mathit{condition} \}$$

It is **read** as the set consisting of every *x* that makes the *condition* true.

Consider example $\{city: big(city)=\text{true } \textbf{and } city \in citiesOfUSA\}$

It is **read** as the set consisting of every *city* that satisfies the condition *big*(*city*)=true **and** *city* ∈ *citiesOfUSA.*

1.3 Meaning of set builder notation

Let A = {*city*: *big*(*city*)=true **and** *city* ∈ *citiesOfUSA*}

It is **read** as the set A consisting of every *city* that satisfies the condition *big*(*city*)=true **and** *city* ∈ *citiesOfUSA*.

Assume we have *citiesOfUSA* = {*lubbock*, *happy*, *post*} and function *big*

x	*big(x)*
lubbock	true
happy	false
post	true

What is set A?

We find **every** *city* that satisfies the condition.

A =

1.3 Meaning of set builder notation

Let A = {*city*: *big*(*city*)=true **and** *city* ∈ *citiesOfUSA*}

It is **read** as the set A consisting of every *city* that satisfies the condition *big*(*city*)=true **and** *city* ∈ *citiesOfUSA*.

Assume we have *citiesOfUSA* = {*lubbock*, *happy*, *post*} and function *big*

x	*big(x)*
lubbock	true
happy	false
post	true

What is set A?

We find **every** *city* that satisfies the condition.

A = {*lubbock*, *post*}

1.3 Meaning of set builder notation

Let A = {*city*: *big(city)*=true **and** *city* ∈ *citiesOfUSA*}

Why A = {*lubbock*, *post*}? We find **every** *city* that satisfies the condition, i.e., make the value of the condition being true.

Every *city*: associate everything to *city*. To check the condition for everything *city* may refer to, we find the value of expressions in the condition

1.3 Meaning of set builder notation

Let A = {*city*: *big*(*city*)=true **and** *city* ∈ *citiesOfUSA*}

Every *city*: associate anything to *city*. To check the condition for everything *city* may refer to, we find the value of expressions in the condition

x	*big(x)*
lubbock	true
happy	false
post	true

city	*big(city)*	*city* ∈ *citiesOfUSA*	*big(city)* and *city* ∈ *citiesOfUSA*
lubbock	?	?	?
happy	?	?	?
post	?	?	?
austin	?	?	?
a table	?	?	?
……			

1.3 Meaning of set builder notation

Let $A = \{city\text{: } big(city)\text{=true } \textbf{and } city \in citiesOfUSA\}$

Every *city*: associate anything to *city*. To check the condition for everything *city* may refer to, we find the value of expressions in the condition

x	*big(x)*
lubbock	true
happy	false
post	true

city	*big(city)*	*city* ∈ *citiesOfUSA*	*big(city)* and *city* ∈ *citiesOfUSA*
lubbock	true	true	true
happy	false	true	false
post	true	true	true
austin	?	false	false
a table	?	false	false
……			

1.3 Meaning of set builder notation

Let $A = \{city\text{: } big(city)\text{=true } \textbf{and } city \in citiesOfUSA\}$

city	*big(city)*	*city ∈ citiesOfUSA*	*big(city) and city ∈ citiesOfUSA*
lubbock	true	true	true
happy	false	true	false
post	true	true	true
austin	?	false	false
a table	?	false	false
……			

Therefore, A = {lubbock, post} i.e., cities make condition true!

1.3 Meaning of set builder notation

Expressions vs **value** (again)

What is the similarity of finding the value for the condition with a given *city* value and finding the value of +(+(2, 3), 5)?

We just need to find values of **all** (sub)expressions patiently to get the value of the final expression (or condition in set builder notation).

big(city)

and			
city	*big(city)*	*city* ∈ *citiesOfUSA*	*city* ∈

citiesOfUSA

1. Representing Sets: SET-BUILDER Notation

1.4 More examples

The set builder notation $\{\ x : condition\}$

condition is an expression whose value is true or false or a number of such expressions combined by logical connectives (and, or, if).

Examples of conditions:

- $x > 10 + z$
- $10x + 20y = 200$
- $father(x) = father(y)$

How do you read $\{x\text{: } father(x) = father(peter)\}$?

Can you describe the above set in everyday English?

1.4 More examples

How do you read {x: $father(x) = father(peter)$}?

Can you describe the set in everyday English?

The set of the siblings of Peter whom he shares a father with.

1.4 More examples

Let the value or output of *weight(p)* be the weight of person *p*. A person is *overweight* if the person's weight is over 300 pounds. Let *P* be a set of people. Use set builder notation to represent the set of people in *P* who are overweight.

1.4 More examples

Let the value or output of *weight(p)* be the weight of person *p*. A person is overweight if the person's weight is over 300 pounds. Let *P* be a set of people. Use set builder notation to represent the set of people in *P* who are overweight.

$$\{p: p \in P \text{ and } weight(p) > 300\}$$

1.4 More examples

Let the value or output of *grader(p)* be the grade of person *p*. Let *P* be the set of people in our class. Use set builder notation to represent the 10th graders of our class.

Write an expression to represent the number of 10th graders in our class.

1.4 More examples

Let the value or output of *grader(p)* be the grade of person *p*. Let *P* be the set of people in our class. Use set builder notation to represent the 10th graders of our class.

$$\{x: grader(x) = 10\}$$

Write an expression to represent the number of 10th graders in our class.

$$|\{x: grader(x) = 10\}|$$

1.4 More examples

- Let *class* be the set of students of our class, and *sport* be the set of all sports.
- The output of the function *likes(s, g)* is *true* if student *s* likes game *g*.
- Write an expression to represent the students in our class who like both basketball and football:

1.4 More examples

- Let *class* be the set of students of our class, and *sport* be the set of all sports.
- The output of the function *likes(s, g)* is *true* if student *s* likes game *g*.
- Write an expression to represent the students in our class who like both basketball and football:

  $\{s: s \in class \text{ and } likes(s, basketball) \text{ and } likes(s, football)\}$

2. Tuples

2.1 Pairs - the simplest tuple

2.1.1 Motivation

- How to represent the location of a house? *(A pair: number, street)*
- How to represent a point in a coordinate system?

2.1 Pairs - the simplest tuple

2.1.1 Motivation

- How to represent the location of a house? *(A pair: number, street)*
- How to represent a point in a coordinate system?

  (A pair: the x-coordinate, the y-coordinate)

2.1 Pairs - the simplest tuple

2.1.1 Motivation

- How to represent the location of a house? *(A pair: number, street)*
- How to represent a point in a coordinate system?

  (A pair: the x-coordinate, the y-coordinate)
- How to represent a seat in a theater?

2.1 Pairs - the simplest tuple

2.1.1 Motivation

We used only simple elements, e.g., a city. However, an element could be complex and we need a notion to represent them - tuple.

- How to represent the location of a house? *(A pair: number, street)*
- How to represent a point in a coordinate system?

  (A pair: the x-coordinate, the y-coordinate)
- How to represent a seat in a theater? *(a pair: row by a letter, seat number)*

at

number 1)

2.1.2 Formal definition of pairs

A **pair**, also called an **ordered pair**, is of the form

$$(A, B)$$

where *A* and *B* can be an expression.

Example. We can associate a name or a variable to a pair:

- *cooperHighSchool = (910, Woodrow Road)*
- *texasTechUniversity = (2500, Broadway Street)*

[JK] added empty page to make the page number match with learning outcome question number

[JK] added empty page to make the page number match with learning outcome question number

2.1.3 Examples of pairs

A person’s name usually consists of first name and last name.

Write an expression using *pair* to represent the name of someone you know

2. Tuples

2.2 Tuples (Motivation)

Sometimes we want to represent more complex objects.

- A name with first, middle and last names: (George, W, Bush)
- Details of a person: (Peter, 16 years old, 5’7”, 130 lb)
- A page in a dictionary: (back, background, bacteria, bad, bail, …)
- A flight route from Lubbock to New York City: (LBB, DFW, JFK)

We call them *tuples*.

2.2 Tuples - formal definition

In general, a tuple can have three elements, called *triple*, four elements, called *quadruple*, or any number of components.

A **tuple** has the form (c_1, c_2, …, c_n), where c_1, ..., c_n can be any *expression*.

For example:

- Detailed information about a person on name, height, age, weight can be represented by tuple

  (Peter, 16, 5’7”, 130)

- Flight route between two LBB and JFK can be represent by tuple

  (LBB, DFW, JFK)

2.3 Examples of Tuples

Assume that 6 of our students got scores of 90, 80, 95, 100, 87 and 80. Are you able to represent these scores as a tuple? If so, write the tuple.

2.3 Examples of Tuples

Assume that 6 of our students got scores of 90, 80, 95, 100, 87 and 80. Are you able to represent these scores as a tuple? If so, write the tuple.

(90, 80, 95, 100, 87, 80)

2. Tuples

2.1 Pairs - the simplest tuple

2.2 Tuples

2.3 Tuples and Indices

2.4 Tuple operations

2.3.1 Tuples and Indices (Motivation)

- *Tuples* can represent complex objects with many components. We need to be able to refer to specific component(s) of a tuple.

- Recall the “detailed information about a person”: (Peter, 16, 5’7). How can we refer to each component of this tuple? A natural way is to use the position of the components in the tuple: the 1st one is name, and the 2nd one is the age, and so on. Each of the numerical positions is called an *index*. In fact, we can also use names to name the positions.

2.3.2 Formal definition

An **index** of a tuple is a number or name to refer to the position of a component in the tuple.

- When the index is a number, it is called a **numbered index**.
- When the index is a name, it is called a **named index**.

For all tuples, we have the natural numbered indices (i.e., 1, 2, 3, …). But for some tuples we prefer named indices. For example, the point (3, 4) on the coordinate plane, we say its *x-coordinate* (a name) is 3, and *y-coordinate* (a name) is 4.

2.3.2 Formal definition

A tuple with named indices is called **named tuple**.

A named tuple (c_1, ..., c_n) is usually represented in the following manner:

$$\begin{matrix} (c_1, & c_2, & \ldots, & c_n) \\ name_1, & name_2, & \ldots, & name_n \end{matrix}$$

We can also simply use a tuple to represent the named indices of a named tuple.

E.g., ($name_1$, $name_2$, …, $name_n$)

Examples

Assuming we use the following named tuple to represent a point

(,)

y-coordinate *x-coordinate*

How to represent a point with *x-coordinate* 10, and *y-coordinate* 20?

Examples

Assuming we use the following named tupe to represent a point

(,)

y-coordinate *x-coordinate*

How to represent a point with *x-coordinate* 10, and *y-coordinate* 20?

(20, 10)

Examples

In a test, we know that Aaron got 90, Bill 95, Cecilia 100 and Dina 88. Assume we use the named tuple

(, , ,)

Aaron, Bill, Cecilia, Dina

or a tuple (Aaron, Bill, Cecilia, Dina) to represent the names. Write the tuple represent the students’ scores above

Examples

In a test, we know that Aaron got 90, Bill 95, Cecilia 100 and Dina 88. Assume we use the named tuple

(, , ,)

Aaron, Bill, Cecilia, Dina

or a tuple (Aaron, Bill, Cecilia, Dina) to represent the names. Write the tuple represent the students’ scores above (90, 95, 100, 88).

Examples

Assume we use the named tuple

(, , ,)

Aaron, Bill, Cecilia, Dina

Assume in another test the scores are (90, 80, 70, 75) using the above named tuple.
What is the score of Bill?

Examples

Assume we use the named tuple

(, , ,)

Aaron, Bill, Cecilia, Dina

Assume in another test the scores are (90, 80, 70, 75) using the above named tuple.

What is the score of Bill? 80

Examples - functions as named tuples

Consider the *big* function on cities

x	*big(x)*
lubbock	true
happy	false
post	true

Represent the function as a named tuple:

Examples - functions as named tuples

Consider the *big* function on cities

x	*big(x)*
lubbock	true
happy	false
post	true

Represent the function as a named tuple:

(true , false , true)

lubbock, happy, post

Examples - functions as names tuples

- Recall the *type* of roller coasters:

rollerCoaster	*type(rollerCoaster)*
Wildfire	Wood
Skyline	Steel
Goliath	Wood
Helix	Steel
Banshee	Steel
Black Hole	Steel

- Write the *type* function above as a named tuple:

Examples - functions as names tuples

- Recall the *type* of roller coasters:

rollerCoaster	*type(rollerCoaster)*
Wildfire	Wood
Skyline	Steel
Goliath	Wood
Helix	Steel
Banshee	Steel
Black Hole	Steel

- Write the *type* function above as a named tuple:

(Wood, Steel, Wood, …)
Wildfire, Skyline, Goliath

2. Tuples

2.1 Pairs - the simplest tuple

2.2 Tuples

2.3 Tuples and Indices

2.4 Tuple operations

2.4. Tuple Operations

2.4.1 Operations on Tuples and Constants

2.4.2 Operations on tuples and tuples

2.4.1 Operations on tuples and Constants (Motivation)

Problem:

Consider 3 kids. John, Wilson, and Eric. John has 1 piece of candy, Wilson has 2, and Eric has 3. After giving 2 pieces to each child, how many pieces does each child have?

2.4.1.1 Operations on tuples and Constants (Motivation)

Lets look at how tuples can be used to represent the information in the problem.

How would you represent the initial candies each kid has using a named tuple

(1, 2, 3)

John, Wilson, Eric

How many total candies do the kids have after each kid is given 2 extra candies ?

(3, 4, 5)

Can you design a function, in intentional form to represent the addition above?

2.4.1.1 Operations on tuples and Constants (Motivation)

Lets look at how tuples can be used to represent the information in the problem.

How would you represent the initial candies each kid has using a named tuple

(1, 2, 3)

John, Wilson, Eric

How many total candies do the kids have after each kid is given 2 extra candies ?

(3, 4, 5)

Can you design a function, in intentional form to represent the addition above?

The output of function *add(candies, newC)*, where *candies* is a tuple (of numbers) and *newC* is a number, is a tuple which is the result of adding *newC* to each component of *candies*.

2.4.1.1 Operations on tuples and Constants (Motivation)

The output of function *add(candies, newC)*, where *candies* is a tuple (of numbers) and *newC* is a number, is a tuple which is the result of adding *newC* to each component of *candies*.

In fact, this function is so useful, we decide to give it a special name "+" to represent the *add(tuple, c)* function, and we use it in the usual infix form. To represent the information of kids obtaining two new candies we use:

2.4.1.1 Operations on tuples and Constants (Motivation)

The output of function *add(candies, newC)*, where *candies* is a tuple (of numbers) and *newC* is a number, is a tuple which is the result of adding *newC* to each component of *candies*.

In fact, this function is so useful, we decide to give it a special name "+" to represent the *add(tuple, c)* function, and we use it in the usual infix form. To represent the information of kids obtaining two new candies we use:

$$(1, 2, 3) + 2$$

2.4.1.1 Operations on tuples and Constants (Motivation)

The output of function *add(candies, newC)*, where *candies* is a tuple (of numbers) and *newC* is a number, is a tuple which is the result of adding *newC* to each component of *candies*.

In fact, this function is so useful, we decide to give it a special name "+" to represent the *add(tuple, c)* function, and we use it in the usual infix form. To represent the information of kids obtaining two new candies we use:

$$(1, 2, 3) + 2$$

The value of the expression is to add 2 to each component $(1+2, 2+2, 3+2) = (3, 4, 5)$

2.4.1.2 Formal definition: tuple + number

Definition (addition of a tuple and a constant): (we offer two definitions here)

- For a tuple A of numbers and a constant c, $A + c$ is the tuple obtained from A by adding c to each of its component.

- Given a tuple $(x_1, x_2, \ldots, x_n)$, where n is a number, and $x_1, \ldots, x_n$ are numbers, and a number c,
  $(x_1, x_2, \ldots, x_n) + c$ is defined as $(x_1+c, x_2+c, \ldots, x_n+c)$

2.4.1.3 Example of tuple + number

$(x_1, x_2, \ldots, x_n) + c$ is defined as $(x_1+c, x_2+c, \ldots, x_n+c)$

Examples to illustrate:

- (10, 21) + 8 = ?
- (1, 2, 3) + 6 = ?

2.4.1.3 Example of tuple + number

$(x_1, x_2, \ldots, x_n) + c$ is defined as $(x_1+c, x_2+c, \ldots, x_n+c)$

Examples to illustrate:

- (10, 21) + 8 = (18, 29)
- (1, 2, 3) + 6 = (7, 8, 9)

2.4.1.3 Application of tuple + number

Problem

Consider 3 kids John, Wilson, and Eric. John's initial pocket money is $10, Wilson's is $8, and Eric's is $12. Their pocket money increases by $3 each. Write an expression to represent the final pocket money of John, Wilson, and Eric using *tuples* and addition (+).

2.4.1.3 Application of tuple + number

Problem

Consider 3 kids John, Wilson, and Eric. John’s initial pocket money is \$10, Wilson’s is \$8, and Eric’s is \$12. Their pocket money increases by \$3 each. Write an expression to represent the final pocket money of John, Wilson, and Eric using *tuples* and addition (+).

(10, 8, 12) + 3

What is the value of the expression?

2.4.1.3 Application of tuple + number

Problem

Consider 3 kids John, Wilson, and Eric. John's initial pocket money is $10, Wilson's is $8, and Eric's is $12. Their pocket money increases by $3 each. Write an expression to represent the final pocket money of John, Wilson, and Eric using tuples and addition.

(10, 8, 12) + 3

What is the value of the expression?

(13, 11, 15)

2.4.1.4 More application of tuple + number

Problem

Consider 3 kids John, Wilson, and Eric. John's initial pocket money is \$10, Wilson's is \$8, and Eric's is \$12. Their pocket money is **doubled**. Are you able to **invent** a new operation (like we did for +) to represent the final pocket money of John, Wilson, and Eric using tuples and multiplication.

2.4.1.4 More application of tuple + number

Problem

Consider 3 kids John, Wilson, and Eric. John's initial pocket money is $10, Wilson's is $8, and Eric's is $12. Their pocket money is doubled. Are you able to invent a new operation (like we did for +) to represent the final pocket money of John, Wilson, and Eric using tuples and multiplication.

(10, 8, 12) * 2

What is the value of the expression?

2.4.1.4 More application of tuple + number

Problem

Consider 3 kids John, Wilson, and Eric. John's initial pocket money is $10, Wilson's is $8, and Eric's is $12. Their pocket money is **doubled**. Write an expression to represent the final pocket money of John, Wilson, and Eric using tuples and multiplication.

(10, 8, 12) * 2

What is the value of the expression?

(20, 16, 24)

2.4.1.4 More application of tuple + number

In addition to *, we can apply other arithmetic operations to a tuple and a number.

Are you able to find other operations over a tuple and a number?

2.4.1.4 More application of tuple + number

In addition to *, we can apply other arithmetic operations to a tuple and a number.

Are you able to find other operations over a tuple and a number?

%, …

2.4.1.5 More application of tuple + number

Recall: $(x_1, x_2, \ldots, x_n)$ + c is defined as $(x_1+c, x_2+c, \ldots, x_n+c)$

Write formal definitions for * (/, %) on a tuple and a number.

$(x_1, x_2, \ldots, x_n) * c$ is

$(x_1, x_2, \ldots, x_n) / c$ is

2.4.1.5 More application of tuple + number

Recall: $(x_1, x_2, \ldots, x_n) + c$ is defined as $(x_1+c, x_2+c, \ldots, x_n+c)$

Write formal definitions for * (/, %) on tuple and number.

$(x_1, x_2, \ldots, x_n) * c$ is defined as: $(x_1 * c, x_2 * c, \ldots, x_n * c)$

$(x_1, x_2, \ldots, x_n) / c$ is defined as: $(x_1 / c, x_2 / c, \ldots, x_n / c)$

2.4. Tuple Operations

2.4.1 Operations on Tuples and Constants

2.4.2 Operations on tuple and tuple

2.4.2.1 Operations on tuple and tuple (motivation)

Problem

In the Halloween Eve, in visiting the first home, Jamie got 1 piece of candy, Aaron got 2, and Eric got 3. For the second home, Jamie got 1 piece of candy, Aaron got 3, and Eric got 2, how many pieces does each child have in total after the two visits?

2.4.2.1 Operations on tuple and tuple (motivation)

Problem

In the Halloween Eve, in visiting the first home, Jamie got 1 piece of candy, Aaron got 2, and Eric got 3. For the second home, Jamie got 1 piece of candy, Aaron got 3, and Eric got 2, how many pieces does each child have in total after the two visits?

Let's look at the problem using tuples:

- Candies got in the 1st visits: (1, 2, 3)
- Candies got in the 2nd visits: (1, 3, 2)

Can you try to **invent** a new operation (as we did before) for getting the total candies? Are you able to define it?

2.4.2.1 Operations on tuple and tuple (motivation)

Can you try to invention a new operation (as we did before) for getting the total candies? Are you able to define it?

We use + to add two tuples.

(1, 2, 3) + (1, 3, 2)

By component-wise addition of the tuples, i.e., (1+1, 2+3, 3+2), we get the value of the expression: (2, 5, 5).

2.4.2.2 Formal definition of operations on tuple and tuple

Definition (addition of a tuple and a tuple): (we offer two definitions here)

- For two tuple A and B of numbers, $A + B$ is the tuple obtained from A and B by adding their corresponding components.

- Given two tuples $(x_1, x_2, \ldots, x_n)$ and $(y_1, y_2, \ldots, y_n)$, where n is a number, and $x_1, \ldots, x_n$, $y_1, \ldots, y_n$ are numbers,
  $(x_1, x_2, \ldots, x_n) + (y_1, y_2, \ldots, y_n)$ is defined as:

2.4.2.2 Formal definition of operations on tuple and tuple

Definition (addition of a tuple and a tuple): (we offer two definitions here)

- For two tuple A and B of numbers, $A + B$ is the tuple obtained from A and B by adding their corresponding components.

- Given two tuples $(x_1, x_2, \ldots, x_n)$ and $(y_1, y_2, \ldots, y_n)$, where n is a number, and $x_1, \ldots, x_n$, $y_1, \ldots, y_n$ are numbers,
  $(x_1, x_2, \ldots, x_n) + (y_1, y_2, \ldots, y_n)$ is defined as $(x_1+y_1, x_2+y_2, \ldots, x_n+y_n)$

2.4.2.3 Example of operations on tuple and tuple

Examples to illustrate:

- (1, 2) + (2, 1) = ?
- (1, 2, 3) + (2, 2, 2) = ?

2.4.2.3 Example of operations on tuple and tuple

Examples to illustrate:

- (1, 2) + (2, 1) = (3, 3)
- (1, 2, 3) + (2, 2, 2) = (3, 4, 5)

2.4.2.4 Application of operations on tuple and tuple

In the first week, John spends $1 for a video game, and Peter spends $2 for a video game. In the second week, John spends $3 for a game, and Peter spends $2 for a game. Write an expression to represent the total cost spent by John and Peter using *tuples* and additions (+).

2.4.2.4 Application of operations on tuple and tuple

In the first week, John spends $1 for a video game, and Peter spends $2 for a video game. In the second week, John spends $3 for a game, and Peter spends $2 for a game. Write an expression to represent the total cost spent by John and Peter using tuples and additions.

$$(1, 2) + (3, 2)$$

What is its value?

2.4.2.4 Application of operations on tuple and tuple

In the first week, John spends $1 for a video game, and Peter spends $2 for a video game. In the second week, John spends $3 for a game, and Peter spends $2 for a game. Write an expression to represent the total cost spent by John and Peter using tuples and additions.

$$(1, 2) + (3, 2)$$

What is its value?

$$(4, 4)$$

2.4.1.6 More application of tuple + number

The prices of 4 products A, B, C and D are $2, $5, $6 and $9 respectively. If there is a tax of 10% on each of these products, write an expression to represent the final costs of A, B, C and D using tuples, + and *. Recall that the final cost of a product is its price plus its price times the tax rate.

2.4.1.6 More application of tuple + number

The prices of 4 products A, B, C and D are $2, $5, $6 and $9 respectively. If there is a tax of 10% on each of these products, write an expression to represent the final costs of A, B, C and D using tuples, + and *. Recall that the final cost of a product is its price plus its price times the tax rate.

(2, 5, 6, 9) + ((2, 5, 6, 9) * 10%)

What is its value? (we do multiplication first)

2.4.1.6 More application of tuple + tuple

The prices of 4 products A, B, C and D are \$2, \$5, \$6 and \$9 respectively. If there is a tax of 10% on each of these products, write an expression to represent the final costs of A, B, C and D using tuples, + and *. Recall that the final cost of a product is its price plus its price times the tax rate.

(2, 5, 6, 9) + ((2, 5, 6, 9) * 10%)

What is its value?
(2.2, 5.5, 6.6., 9.9)

Let *x* be tuple (2, 5, 6, 9), can you rewrite the expression above using *x*?

2.4.2.6 More application of operations on tuple and tuple

Following the idea of addition of tuple and tuple, write formal definitions for * (/, %) on tuple and tuple.

$(x_1, x_2, \ldots, x_n)$ ***** $(y_1, y_2, \ldots, y_n)$ is defined as:

$(x_1, x_2, \ldots, x_n)$ / $(y_1, y_2, \ldots, y_n)$ is defined as:

2.4.2.6 More application of operations on tuple and tuple

Following the idea of addition of tuple and tuple, write formal definitions for * (/, %) on tuple and tuple.

$(x_1, x_2, \ldots, x_n)$ ***** $(y_1, y_2, \ldots, y_n)$ is defined as: $(x_1 * y_1, x_2 * y_2, \ldots, x_n * y_n)$

$(x_1, x_2, \ldots, x_n)$ / $(y_1, y_2, \ldots, y_n)$ is defined as: $(x_1 / y_1, x_2 / y_2, \ldots, x_n / y_n)$

Summary

- We have learned to use set builder notation to represent sets we are interested in.
- We have learned
  - Tuples, their components and operations (functions) over tuples
  - How to use tuples and operations over them to represent knowledge in a problem