CS250: Discrete Math for Computer Science

L14: Division and Modular Arithmetic

Division Algorithm and Modular Arithmetic

Thm $\forall n\, d\, (d > 0 \to \exists! q \exists! r\, (n = q \cdot d + r\ \wedge\ 0 \le r < d))$

Division Algorithm and Modular Arithmetic

Thm $\forall n\, d\ (d > 0 \rightarrow \exists! q \exists! r\ (n = q \cdot d + r \ \wedge\ 0 \leq r < d))$

Notation: For n, d as above $\quad n \operatorname{div} d \stackrel{\text{def}}{=} q \quad n \% d \stackrel{\text{def}}{=} r$

Division Algorithm and Modular Arithmetic

Thm $\forall n\, d\, (d > 0 \to \exists! q \exists! r\, (n = q \cdot d + r \ \wedge\ 0 \le r < d))$

Notation: For n, d as above $\quad n \operatorname{div} d \overset{\text{def}}{=} q \quad n \% d \overset{\text{def}}{=} r$

$$
\begin{array}{rrcl@{\qquad}rr@{\qquad}rr}
n = & q \cdot d & + & r & q = & n \operatorname{div} d & r = & n \% \ d \\
3 = & 0 \cdot 10 & + & 3 & 0 = & 3 \operatorname{div} 10 & 3 = & 3 \% 10 \\
21 = & 2 \cdot 10 & + & 1 & 2 = & 21 \operatorname{div} 10 & 1 = & 21 \% 10 \\
128 = & 12 \cdot 10 & + & 8 & 12 = & 128 \operatorname{div} 10 & 8 = & 128 \% 10 \\
-7 = & -1 \cdot 10 & + & 3 & -1 = & -7 \operatorname{div} 10 & 3 = & -7 \% 10 \\
3 = & 1 \cdot 2 & + & 1 & 1 = & 3 \operatorname{div} 2 & 1 = & 3 \% \ 2 \\
21 = & 10 \cdot 2 & + & 1 & 10 = & 21 \operatorname{div} 2 & 1 = & 21 \% \ 2 \\
128 = & 64 \cdot 2 & + & 0 & 64 = & 128 \operatorname{div} 2 & 0 = & 128 \% \ 2 \\
-7 = & -4 \cdot 2 & + & 1 & -4 = & -7 \operatorname{div} 2 & 1 = & -7 \% \ 2
\end{array}
$$

iClicker 14.1 What is $(-3)\%2$

A: -1 **B: 0** **C: 1**

iClicker 14.1 What is $(-3)\%2$

A: -1 B: 0 C: 1

$$-3 = -2 \cdot 2 + 1$$

iClicker 14.2 What is $(-3)\%4$

A: -1 **B: 0** **C: 1** **D: 3**

iClicker 14.2 What is $(-3)\%4$

A: -1 **B: 0** **C: 1** **D: 3**

$$-3 \quad = \quad -1 \cdot 4 + 1$$

Abbreviations

See our list of abbreviations:

`people.cs.umass.edu/~immerman/cs250/Notation250.pdf`

$\hookrightarrow$ is an abbreviation for "is an abbreviation for"

Abbreviations

See our list of abbreviations:

`people.cs.umass.edu/~immerman/cs250/Notation250.pdf`

$\hookrightarrow$ is an abbreviation for "is an abbreviation for"

$$
\begin{array}{rcl}
t_1 \neq t_2 & \hookrightarrow & \sim(t_1 = t_2) \\
(\forall x \,.\, \alpha)\beta & \hookrightarrow & \forall x(\alpha \to \beta) \\
(\exists x \,.\, \alpha)\beta & \hookrightarrow & \exists x(\alpha \wedge \beta) \\
\exists! x(\alpha(x)) & \hookrightarrow & \exists x \forall y(\alpha(x) \wedge (\alpha(y) \to y = x)) \\
x < y & \hookrightarrow & x \leq y \ \wedge \ x \neq y \\
x | y & \hookrightarrow & \exists z \ (x \cdot z = y) \\
\mathrm{prime}(x) & \hookrightarrow & 1 < x \ \wedge \ \forall y \ (1 < y \wedge y | x \ \to \ y = x)
\end{array}
$$

Abbreviations

See our list of abbreviations:

`people.cs.umass.edu/~immerman/cs250/Notation250.pdf`

$$
\begin{array}{rcl}
\hookrightarrow & \text{is an abbreviation for} & \text{“is an abbreviation for”} \\
t_1 \neq t_2 & \hookrightarrow & \sim(t_1 = t_2) \\
(\forall x \,.\, \alpha)\beta & \hookrightarrow & \forall x(\alpha \to \beta) \\
(\exists x \,.\, \alpha)\beta & \hookrightarrow & \exists x(\alpha \wedge \beta) \\
\exists! x(\alpha(x)) & \hookrightarrow & \exists x \forall y(\alpha(x) \wedge (\alpha(y) \to y = x)) \\
x < y & \hookrightarrow & x \leq y \ \wedge \ x \neq y \\
x | y & \hookrightarrow & \exists z \ (x \cdot z = y) \\
\mathrm{prime}(x) & \hookrightarrow & 1 < x \ \wedge \ \forall y \ (1 < y \wedge y | x \ \to \ y = x)
\end{array}
$$

$$\text{Primes} \quad = \quad \{2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, \ldots\}$$

Abbreviations

See our list of abbreviations:

`people.cs.umass.edu/~immerman/cs250/Notation250.pdf`

$$
\begin{array}{rcl}
\hookrightarrow & \text{is an abbreviation for} & \text{“is an abbreviation for”} \\
t_1 \neq t_2 & \hookrightarrow & \sim(t_1 = t_2) \\
(\forall x \,.\, \alpha)\beta & \hookrightarrow & \forall x(\alpha \to \beta) \\
(\exists x \,.\, \alpha)\beta & \hookrightarrow & \exists x(\alpha \wedge \beta) \\
\exists ! x(\alpha(x)) & \hookrightarrow & \exists x \forall y(\alpha(x) \wedge (\alpha(y) \to y = x)) \\
x < y & \hookrightarrow & x \leq y \;\wedge\; x \neq y \\
x|y & \hookrightarrow & \exists z\,(x \cdot z = y) \\
\mathrm{prime}(x) & \hookrightarrow & 1 < x \;\wedge\; \forall y\,(1 < y \wedge y|x \;\to\; y = x)
\end{array}
$$

$$
\begin{array}{rcl}
\text{Primes} & = & \{2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, \ldots\} \\
& = & \left\{ a \in \mathbf{Z} \;\middle|\; \mathbf{Z}[a/x] \models \mathrm{prime}(x) \right\}
\end{array}
$$

Congruence mod m

Def For $m > 1$ define the **congruence mod m** relation on **Z** as follows: $a \equiv b \pmod{m}$ iff $m|(a-b)$

Congruence mod m

Def For $m > 1$ define the **congruence mod *m*** relation on **Z** as follows: $a \equiv b \pmod{m}$ iff $m | (a - b)$

Examples

$3 \equiv 13 \pmod{10}$

Congruence mod m

Def For $m > 1$ define the **congruence mod m** relation on **Z** as follows: $a \equiv b \pmod{m}$ iff $m|(a-b)$

Examples

$$3 \equiv 13 \pmod{10}$$
$$3 \equiv -7 \pmod{10}$$

Congruence mod m

Def For $m > 1$ define the **congruence mod *m*** relation on **Z** as follows: $a \equiv b \pmod{m}$ iff $m|(a-b)$

Examples

$$3 \equiv 13 \pmod{10}$$
$$3 \equiv -7 \pmod{10}$$
$$128 \equiv 13 \pmod{5}$$

Congruence mod m

Def For $m > 1$ define the **congruence mod m** relation on **Z** as follows: $a \equiv b \pmod{m}$ iff $m|(a-b)$

Examples

$$3 \equiv 13 \pmod{10}$$
$$3 \equiv -7 \pmod{10}$$
$$128 \equiv 13 \pmod{5}$$
$$128 \equiv 0 \pmod{2}$$

Congruence mod m

Def For $m > 1$ define the **congruence mod *m*** relation on **Z** as follows: $a \equiv b \pmod{m}$ iff $m | (a - b)$

Examples

$$3 \equiv 13 \pmod{10}$$
$$3 \equiv -7 \pmod{10}$$
$$128 \equiv 13 \pmod{5}$$
$$128 \equiv 0 \pmod{2}$$

iClicker 14.3 True or False: $3 \equiv -2 \pmod{5}$

A: True **B: False**

Congruence mod m

Def For $m > 1$ define the **congruence mod m** relation on **Z** as follows: $a \equiv b \pmod{m}$ iff $m | (a - b)$

Examples

$$3 \equiv 13 \pmod{10}$$

$$3 \equiv -7 \pmod{10}$$

Congruence mod m

Def For $m > 1$ define the **congruence mod m** relation on **Z** as follows: $a \equiv b \pmod{m}$ iff $m | (a - b)$

Examples

$$3 \equiv 13 \pmod{10}$$

$$3 \equiv -7 \pmod{10}$$

$$128 \equiv 13 \pmod{5}$$

Congruence mod m

Def For $m > 1$ define the **congruence mod m** relation on **Z** as follows: $a \equiv b \pmod{m}$ iff $m|(a-b)$

Examples

$$3 \equiv 13 \pmod{10}$$
$$3 \equiv -7 \pmod{10}$$
$$128 \equiv 13 \pmod{5}$$
$$128 \equiv 0 \pmod{2}$$

Congruence mod m

Def For $m > 1$ define the **congruence mod m** relation on **Z** as follows: $a \equiv b \pmod{m}$ iff $m|(a-b)$

Examples

$$
\begin{aligned}
3 &\equiv 13 \pmod{10} \\
3 &\equiv -7 \pmod{10} \\
128 &\equiv 13 \pmod{5} \\
128 &\equiv 0 \pmod{2}
\end{aligned}
$$

Prop $(\forall a\, b\, m\, .\, m > 1)(a \equiv b \pmod{m} \leftrightarrow a\%m = b\%m)$

Congruence mod m

Def For $m > 1$ define the **congruence mod m** relation on **Z** as follows: $a \equiv b \pmod{m}$ iff $m|(a-b)$

Examples

$$3 \equiv 13 \pmod{10}$$
$$3 \equiv -7 \pmod{10}$$
$$128 \equiv 13 \pmod{5}$$
$$128 \equiv 0 \pmod{2}$$

Prop $(\forall a\, b\, m \,.\, m > 1)(a \equiv b \pmod{m} \leftrightarrow a\%m = b\%m)$

Observation

$x \equiv 0 \pmod{2}$ iff x is even

$x \equiv 1 \pmod{2}$ iff x is odd.

$a \equiv 0 \pmod{m}$ iff $m|a$

Fourth grade trick for checking multiplication.

$25\%9 = 7$ $289\%9 = 1$ $7225\%9 = 7$

Casting Out Nines

Fourth grade trick for checking multiplication.

After multiplying two numbers, here's a check that you are right:

$$25 \quad \times \quad 289 \quad = \quad 7225$$

$25\%9 = 7$ $\qquad$ $289\%9 = 1$ $\qquad$ $7225\%9 = 7$

Casting Out Nines

Fourth grade trick for checking multiplication.

After multiplying two numbers, here's a check that you are right:

Sum the digits and multiply the sums of digits

$$\begin{array}{ccccc} 25 & \times & 289 & = & 7225 \\ & & 19 \to 10 & & 16 \\ 7 & \times & 1 & = & 7 \end{array}$$

$25\%9 = 7$ $\quad$ $289\%9 = 1$ $\quad$ $7225\%9 = 7$

Casting Out Nines

Fourth grade trick for checking multiplication.

After multiplying two numbers, here's a check that you are right:

Sum the digits and multiply the sums of digits

$$\begin{array}{ccccc} 25 & \times & 289 & = & 7225 \\ & & 19 \to 10 & & 16 \\ 7 & \times & 1 & = & 7 \end{array}$$

Why does this work?

$25\%9 = 7$ $\quad$ $289\%9 = 1$ $\quad$ $7225\%9 = 7$

Casting Out Nines

Fourth grade trick for checking multiplication.

After multiplying two numbers, here's a check that you are right:

Sum the digits and multiply the sums of digits

$$
\begin{array}{ccccc}
25 & \times & 289 & = & 7225 \\
 & & 19 \to 10 & & 16 \\
7 & \times & 1 & = & 7
\end{array}
$$

Why does this work?

Reducing mod 9 is the same as summing the digits.

$25\%9 = 7 \qquad 289\%9 = 1 \qquad 7225\%9 = 7$

Completeness and Soundness for Natural Deduction

$$\text{PredCalcSAT} \stackrel{\text{def}}{=} \{\varphi \in \text{PredCalc} \mid \text{there exists } W,\ W \models \varphi\}$$

$$\text{PredCalcVALID} \stackrel{\text{def}}{=} \{\varphi \in \text{PredCalc} \mid \text{for all } W,\ W \models \varphi\}$$

Prop For all $\varphi \in \text{PredCalc}$,

$$\varphi \in \text{PredCalcSAT} \quad \text{iff} \quad \sim\varphi \notin \text{PredCalcVALID}$$

$$\varphi \in \text{PredCalcVALID} \quad \text{iff} \quad \sim\varphi \notin \text{PredCalcSAT}$$

Soundness Theorem for Natural Deduction For all $\varphi \in \text{PredCalc}$, if $\vdash \varphi$ then $\varphi \in \text{PredCalcVALID}$.

Completeness Theorem for Natural Deduction [Gödel's Ph.D. thesis, 1929] For all $\varphi \in \text{PredCalc}$,
if $\varphi \in \text{PredCalcVALID}$ then $\vdash \varphi$.

A formula of PredCalc is provable by Natural Deduction iff it is true in all worlds.