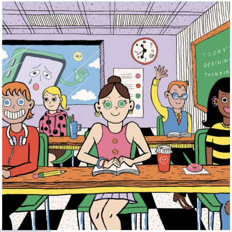
전자기기 사용 금지

Here's What Happened When I Made My College Students Put Away Their Phones

Aug. 21, 2025



Computers & Education

Volume 62, March 2013, Pages 24-31

Laptop multitasking hinders classroom learning for both users and nearby peers

Faria Sana [a], Tina Weston [b c], Nicholas J. Cepeda [b c]

Show more

Outline | Share | Cite

https://doi.org/10.1016/j.compedu.2012.10.003

Get rights and content

Open access

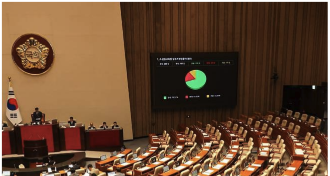
Abstract

Laptops are commonplace in university classrooms. In light of cognitive psychology theory on costs associated with multitasking, we examined the effects of in-class laptop use on student learning in a simulated classroom. We found that

한국경제

내년 3월부터 수업 중 스마트폰 쓰면 '불법'...학생들 '멘붕'

이슬기 · 2025. 8. 27. 16:21



SNU 프로그래밍언어 특강

(0.0)

이 광근

kwangkeunyi.snu.ac.kr

인덕$_{\text{induction},\text{귀납}}$ = 집합의 정의

집합을 정의하는 방법

- 인덕$_{\text{induction},\text{귀납}}$: 만든것으로 만들기
- 그 집합의 원소를 가지고 그 집합의 원소를 만든다
- 유한개의 그런 규칙들로 표현

그 집합은 이것이다

- 규칙 $\frac{X}{x}$: 가정 X와 결론 x.
- "X가 정의하려는 집합에 모두 있으면, x도 있어야 한다."
- 그러한 집합 중에서 가장 작은 집합.
  - 규칙 $\frac{X}{x}$들 집합을 Φ라고 하자
  - 규칙들 Φ는 함수 ϕ를 정의:

$$\phi(Y) = \{x \mid \frac{X}{x} \in \Phi,\ X \subseteq Y\}$$

  - Φ 규칙들이 정의하는 집합은 함수 ϕ에 의해서 닫혀있는 모든 집합들의 교집합

$$\bigcap\{X | \phi(X) \subseteq X\}$$

자연수의 집합:

$$(\emptyset, 0) \quad (\{n\}, n+1)$$

영문 소문자 알파벳으로 만들어 지는 스트링의 집합:

$$(\emptyset, \epsilon) \quad (\{\alpha\}, x\alpha \text{ for } x \in \{\mathtt{a}, \cdots, \mathtt{z}\})$$

인덕$_{\text{induction}}$규칙 표기법

자연수 집합은

$$n \;\to\; 0 \mid n+1$$

혹은

$$\frac{}{0} \qquad \frac{n}{n+1}$$

위의 스트링 집합은:

$$\begin{array}{rcll} \alpha & \to & \epsilon & \\ & \mid & x\alpha & (x \in \{\mathtt{a}, \cdots, \mathtt{z}\}) \end{array}$$

혹은

$$\frac{}{\epsilon} \qquad \frac{\alpha}{x\alpha}\; x \in \{\mathtt{a}, \cdots, \mathtt{z}\}$$

집합이 유한하다면 인덕이 필요없다. 집합 $\{1, 2, 3\}$을 규칙들로 표현하면

$$(\emptyset, 1) \quad (\emptyset, 2) \quad (\emptyset, 3)$$

혹은

$$x \to 1 \mid 2 \mid 3$$

혹은

$$\frac{}{1} \quad \frac{}{2} \quad \frac{}{3}$$

가 된다.

리스트의 집합:

$$\frac{}{\texttt{nil}} \qquad \frac{\ell}{\circ\!-\!\ell}$$

혹은

$$\ell \;\rightarrow\; \texttt{nil} \mid \circ\!-\!\ell$$

말단에 정수를 가지는 두갈래 나무$_{\text{binary tree}}$들의 집합:

$$\frac{}{n}\ n \in \mathbb{Z} \qquad \frac{t}{\mathtt{N}(t, \mathtt{nil})}$$

$$\frac{t}{\mathtt{N}(\mathtt{nil}, t)} \qquad \frac{t_1 \quad t_2}{\mathtt{N}(t_1, t_2)}$$

혹은

$$\begin{array}{rcll} t & \to & n & (n \in \mathbb{Z}) \\ & | & \mathtt{N}(t, \mathtt{nil}) & \\ & | & \mathtt{N}(\mathtt{nil}, t) & \\ & | & \mathtt{N}(t, t) & \end{array}$$

정수식들의 집합:

$$\frac{}{n}\; n \in \mathbb{N} \qquad \frac{e}{-e}$$

$$\frac{e_1 \quad e_2}{e_1 + e_2} \qquad \frac{e_1 \quad e_2}{e_1 * e_2}$$

혹은

$$\begin{array}{rcll} e & \rightarrow & n & (n \in \mathbb{N}) \\ & | & -e & \\ & | & e + e & \\ & | & e * e & \end{array}$$

그 집합은 이렇게 만든다

이 집합

$$\bigcap\{X|\phi(X) \subseteq X\}$$

을 이렇게 만들 수 있다:

$$\bigcup_{i\in\mathbb{N}} \phi^i(\emptyset) = \phi^0(\emptyset) \cup \phi^1(\emptyset) \cup \phi^2(\emptyset) \cup \cdots$$

유한한 크기($i \in \mathbb{N}$)의 것들이 모인 무한한 집합.

자연수 집합 규칙

$$n \;\to\; 0 \mid n+1$$

그 집합은:

$$\begin{array}{rcl} \phi^0(\emptyset) & = & \emptyset \\ \phi^1(\emptyset) & = & \{0\} \\ \phi^2(\emptyset) & = & \{0,1\} \\ \phi^3(\emptyset) & = & \{0,1,2\} \\ \ldots & & \end{array}$$

들의 합집합.

리스트의 집합 규칙

$$\ell \ \rightarrow \ \texttt{nil} \mid \circ\!-\!\ell$$

그 집합은:

$$\begin{array}{rcl}
\phi^0(\emptyset) & = & \emptyset \\
\phi^1(\emptyset) & = & \{\texttt{nil}\} \\
\phi^2(\emptyset) & = & \{\texttt{nil}, \circ-\texttt{nil}\} \\
\phi^3(\emptyset) & = & \{\texttt{nil}, \circ-\texttt{nil}, \circ-\circ-\texttt{nil}\} \\
\ldots & &
\end{array}$$

들의 합집합.

두갈래 나무$_{\text{binary tree}}$의 집합 규칙:

$$\begin{array}{rcl} t & \to & \circ \\ & | & \mathtt{N}(t, \mathtt{nil}) \\ & | & \mathtt{N}(\mathtt{nil}, t) \\ & | & \mathtt{N}(t, t) \end{array}$$

이 집합은:

$$\begin{array}{rcl} \phi^0(\emptyset) & = & \emptyset \\ \phi^1(\emptyset) & = & \{\circ\} \\ \phi^2(\emptyset) & = & \{\circ, \mathtt{N}(\circ, \mathtt{nil}), \mathtt{N}(\mathtt{nil}, \circ), \mathtt{N}(\circ, \circ)\} \\ \dots & & \end{array}$$

들의 합집합.

스트링 집합 규칙:

$$\begin{array}{rcll} \alpha & \to & \epsilon & \\ & | & x\alpha & (x \in \{\mathtt{a}, \cdots, \mathtt{z}\}) \end{array}$$

이 집합은

$$\begin{array}{rcl} \phi^0(\emptyset) & = & \emptyset \\ \phi^1(\emptyset) & = & \{\epsilon\} \\ \phi^2(\emptyset) & = & \{\epsilon, \mathtt{a}\epsilon, \cdots, \mathtt{z}\epsilon\} \\ \ldots & & \end{array}$$

들의 합집합.

밑바닥 없는 규칙

집합 규칙:

$$\begin{array}{rcl} t & \to & \texttt{N}(t, \texttt{nil}) \\ & \mid & \texttt{N}(\texttt{nil}, t) \\ & \mid & \texttt{N}(t, t) \end{array}$$

이 집합은 $\emptyset$:

$$\phi^0(\emptyset) = \phi^1(\emptyset) = \cdots = \emptyset$$

모든 귀납 규칙

$$\frac{X}{x}$$

에서 $X \neq \emptyset$이면 정의하는 집합은 $\emptyset$

정리

- 인덕$_{\text{induction,귀납}}$ = 집합의 정의(inductive definition)
- 인덕규칙들 Φ 표현 방법들
- 그 집합은 무엇이지?
  - "닫혀있는 최소의 집합"
  - $\bigcap\{A|\phi(A) \subseteq A\}$
- 그 집합은 어떻게 만들지?
  - $\emptyset$에서 출발해서, 규칙을 유한번 적용해서 만드는 원소들만을 빠짐없이 첨가

$$\bigcup_{i \in \mathbb{N}} \phi^i(\emptyset)$$

참고: 코덕$_{\text{coinduction}}$ = 무한원소 포함한 집합정의

- 코덕$_{\text{coinduction}}$으로 정의하는 집합
  = 규칙들 Φ을 존중한다면 모두 포함시키기
- 규칙들 Φ가 정의하는 함수 ϕ:

$$\phi(Y) = \{x \mid \frac{X}{x} \in \Phi,\ X \subseteq Y\}$$

- 코덕으로 정의하는 집합은 함수 ϕ가 확장하는 모든 집합들의 합집합

$$\bigcup\{X | X \subseteq \phi(X)\}$$

- 이 집합은 이렇게 만든다: ($\mathbb{U}$는 무한원소를 포함한 전체집합)

$$\bigcap_{i \in \mathbb{N}} \phi^i(\mathbb{U}) = \phi^0(\mathbb{U}) \cap \phi^1(\mathbb{U}) \cap \phi^2(\mathbb{U}) \cap \cdots$$

  무한원소도 포함하는 집합

참고: 정의하는 집합 두 가지

인덕$_{\text{induction}}$으로 정의하면	코덕$_{\text{coinduction}}$으로 정의하면
$\bigcap\{X \| \phi(X) \subseteq X\}$	$\bigcup\{X \| X \subseteq \phi(X)\}$
$\bigcup_{i\in\mathbb{N}} \phi^i(\emptyset)$	$\bigcap_{i\in\mathbb{N}} \phi^i(\mathbb{U})$
ϕ의 최소고정점	ϕ의 최대고정점

ϕ는 유한개의 규칙들 Φ가 정의하는 함수:

$\phi(Y) = \{x \mid \frac{X}{x} \in \Phi,\ X \subseteq Y\}$

원소들의 순서

인덕$_{\text{induction,귀납법}}$으로 정의된 집합 S

- ϕ^i번째에 새롭게 만들어 지는 원소: i째번 원소.
- 0째번에 만들어진 원소들($\emptyset$)이 씨.
- "바닥있는 순서$_{\text{well-founded order}}$"
- 인덕$_{\text{induction,귀납법}}$으로 정의된 집합은 바닥있는 순서를 가지고 있다.

인덕$_{\text{induction,귀납법}}$ = 증명의 방법

증명 목표:

$$\forall x \in S.P(x)$$

- S가 인덕으로 정의됨, 즉, 모든 원소들의 순서가 있음.
- P(0째번 원소)를 증명: 항상 성립($\emptyset$이므로)
- 임의의 $i > 0$에 대해서

$$(\forall j < i.P(j\text{째번 원소})) \Rightarrow P(i\text{째번 원소})$$

를 증명.

("대우법 증명"도 있음)

증명: 규칙들에 대한 것으로

즉,

- 시작(base case): 1째번 원소
  규칙 $\frac{}{x}$가 만드는 x들에 대해, $P(x)$를 증명.
- 인덕(inductive case): 모든 $1 < k$째번 원소
  규칙 $\frac{X}{x}$가 만드는 x들에 대해, $(\forall a \in X.P(a)) \Rightarrow P(x)$를 증명.

증명 예

- 증명: $\forall n \in \mathbb{N}. 0 + 1 + 2 + \cdots + n = n(n+1)/2$
- 증명: 갈라지면 항상 두갈래로 갈라지는 나무는 말단 노드의 갯수가 내부 노드의 갯수보다 하나 많다.

$$\text{Tree} \quad t \to \circ \mid \mathtt{N}(t, t)$$

$$\forall t \in \text{Tree}. |\text{말단노드}(t)| = |\text{내부노드}(t)| + 1$$

논리식 집합

$$
\begin{array}{rcl}
f & \rightarrow & T \mid F \\
& \mid & \neg f \\
& \mid & f \wedge f \\
& \mid & f \vee f \\
& \mid & f \Rightarrow f
\end{array}
$$

논리식 의미

조립식 정의 compositional definition

$$\begin{array}{rcl}
[\![T]\!] & = & \mathsf{true} \\
[\![F]\!] & = & \mathsf{false} \\
[\![\neg f]\!] & = & \mathsf{not}[\![f]\!] \\
[\![f_1 \land f_2]\!] & = & [\![f_1]\!] \ \mathsf{andalso} \ [\![f_2]\!] \\
[\![f_1 \lor f_2]\!] & = & [\![f_1]\!] \ \mathsf{orelse} \ [\![f_2]\!] \\
[\![f_1 \Rightarrow f_2]\!] & = & [\![f_1]\!] \ \mathsf{implies} \ [\![f_2]\!]
\end{array}$$

임의의 논리식 f의 의미가 정의 된 셈.

$$
\begin{aligned}
&[\![(T \wedge (T \vee F)) \Rightarrow F]\!] \\
= \quad & [\![T \wedge (T \vee F)]\!] \text{ implies } [\![F]\!] \\
= \quad & ([\![T]\!] \text{ andalso } [\![T \vee F]\!]) \text{ implies false} \\
= \quad & (\text{true andalso } ([\![T]\!] \text{ orelse } [\![F]\!])) \text{ implies false} \\
= \quad & (\text{true andalso } (\text{true orelse false})) \text{ implies false} \\
= \quad & \text{false}
\end{aligned}
$$

어떤 집합의 정의

쌍 $(\{g_1, \cdots, g_n\}, f)$들의 집합

$$\frac{}{(\Gamma, T)} \qquad \frac{}{(\Gamma, f)}\ f \in \Gamma \qquad \frac{(\Gamma, F)}{(\Gamma, f)} \qquad \frac{(\Gamma, \neg\neg f)}{(\Gamma, f)}$$

$$\frac{(\Gamma, f_1) \quad (\Gamma, f_2)}{(\Gamma, f_1 \wedge f_2)} \qquad \frac{(\Gamma, f_1 \wedge f_2)}{(\Gamma, f_1)}$$

$$\frac{(\Gamma, f_1)}{(\Gamma, f_1 \vee f_2)} \qquad \frac{(\Gamma, f_1 \vee f_2) \quad (\Gamma \cup \{f_1\}, f_3) \quad (\Gamma \cup \{f_2\}, f_3)}{(\Gamma, f_3)}$$

$$\frac{(\Gamma \cup \{f_1\}, f_2)}{(\Gamma, f_1 \Rightarrow f_2)} \qquad \frac{(\Gamma, f_1 \Rightarrow f_2) \quad (\Gamma, f_1)}{(\Gamma, f_2)}$$

$$\frac{(\Gamma \cup \{f\}, F)}{(\Gamma, \neg f)} \qquad \frac{(\Gamma, f) \quad (\Gamma, \neg f)}{(\Gamma, F)}$$

형식논리의 표기법으로

$$\frac{}{\Gamma \vdash T} \qquad \frac{}{\Gamma \vdash f}\ f \in \Gamma \qquad\qquad \frac{\Gamma \vdash F}{\Gamma \vdash f} \qquad \frac{\Gamma \vdash \neg\neg f}{\Gamma \vdash f}$$

$$\frac{\Gamma \vdash f_1 \quad \Gamma \vdash f_2}{\Gamma \vdash f_1 \wedge f_2} \qquad\qquad \frac{\Gamma \vdash f_1 \wedge f_2}{\Gamma \vdash f_1}$$

$$\frac{\Gamma \vdash f_1}{\Gamma \vdash f_1 \vee f_2} \qquad \frac{\Gamma \vdash f_1 \vee f_2 \quad \Gamma \cup \{f_1\} \vdash f_3 \quad \Gamma \cup \{f_2\} \vdash f_3}{\Gamma \vdash f_3}$$

$$\frac{\Gamma \cup \{f_1\} \vdash f_2}{\Gamma \vdash f_1 \Rightarrow f_2} \qquad\qquad \frac{\Gamma \vdash f_1 \Rightarrow f_2 \quad \Gamma \vdash f_1}{\Gamma \vdash f_2}$$

$$\frac{\Gamma \cup \{f\} \vdash F}{\Gamma \vdash \neg f} \qquad\qquad \frac{\Gamma \vdash f \quad \Gamma \vdash \neg f}{\Gamma \vdash F}$$

또 다른 시선: 증명들의 집합을 정의

증명들의 집합을 만드는 증명규칙$_{\text{inference rules, proof rules}}$

- 예를 들어, 증명규칙

$$\frac{\Gamma \vdash f_1 \quad \Gamma \vdash f_2}{\Gamma \vdash f_1 \land f_2}$$

  은 증명을 만드는 규칙
- $\Gamma \vdash f_1$와 $\Gamma \vdash f_2$의 증명들을 가지고 $\Gamma \vdash f_1 \land f_2$의 증명을 만든다.

증명 나무

$$
\dfrac{\dfrac{\dfrac{}{\{p \to \neg p, p\} \vdash p} \qquad \dfrac{\dfrac{}{\{p \to \neg p, p\} \vdash p \to \neg p} \qquad \dfrac{}{\{p \to \neg p, p\} \vdash p}}{\{p \to \neg p, p\} \vdash \neg p}}{\{p \to \neg p, p\} \vdash F}}{\{p \to \neg p\} \vdash \neg p}
$$

증명 규칙의 평가

기계가 만드는 $\{g_1, \cdots, g_n\} \vdash f$ 는 어떤 것들인가?
예) $[\![g_1 \wedge \cdots \wedge g_n \Rightarrow f]\!] = \mathsf{true}$ 인가?

- 기계의 안전성$_{\mathsf{soundness}}$: 믿을만하다

$$\Gamma \vdash f \quad \text{이면} \quad [\![\Gamma \Rightarrow f]\!] = \mathsf{true}$$

- 기계의 완전성$_{\mathsf{completeness}}$: 빼먹지않는다

$$\Gamma \vdash f \quad \text{면이} \quad [\![\Gamma \Rightarrow f]\!] = \mathsf{true}$$