CTL model checking

EECS 4315

www.eecs.yorku.ca/course/4315/

Course evaluation

The course evaluation can be completed here.

CTL

Definition

The *formulas* are defined by

$$f ::= a \mid f \wedge f \mid \neg f \mid \exists \bigcirc f \mid \exists (f \; \mathsf{U} \; f) \mid \forall \bigcirc f \mid \forall (f \; \mathsf{U} \; f)$$

Semantics of CTL

$$
\begin{array}{rcl}
s \models a & \text{iff} & a \in \ell(s) \\
s \models f_1 \wedge f_2 & \text{iff} & s \models f_1 \text{ and } s \models f_2 \\
s \models \neg f & \text{iff} & \text{not}(s \models f) \\
s \models \exists\bigcirc f & \text{iff} & \exists p \in \mathit{Paths}(s) : p[1] \models f \\
s \models \exists(f_1 \, \mathsf{U} \, f_2) & \text{iff} & \exists p \in \mathit{Paths}(s) : \\
 & & \exists i \geq 0 : p[i] \models f_2 \text{ and } \forall 0 \leq j < i : p[j] \models f_1 \\
s \models \forall\bigcirc f & \text{iff} & \forall p \in \mathit{Paths}(s) : p[1] \models f \\
s \models \forall(f_1 \, \mathsf{U} \, f_2) & \text{iff} & \forall p \in \mathit{Paths}(s) : \\
 & & \exists i \geq 0 : p[i] \models f_2 \text{ and } \forall 0 \leq j < i : p[j] \models f_1
\end{array}
$$

Example

Question

How to express "Each red light is preceded by a green light" in CTL?

Answer

$\neg red \wedge \forall\Box(green \vee \forall\bigcirc\neg red)$

Example

Question

How to express “The light is infinitely often green” in CTL?

Example

Question

How to express "The light is infinitely often green" in CTL?

Answer

$\forall\Box\forall\Diamond$green

Question

Recall that

$$\exists\Diamond f = \exists(\text{true}\ \mathsf{U}\ f).$$

How is

$$s \models \exists\Diamond f$$

defined?

Semantics of CTL

Question

Recall that

$$\exists\Diamond f = \exists(\text{true U } f).$$

How is

$$s \models \exists\Diamond f$$

defined?

Answer

$$\exists p \in \mathit{Paths}(s) : \exists i \geq 0 : p[i] \models f.$$

Question

Recall that

$$\forall \Diamond f = \forall (\text{true} \, \mathsf{U} \, f)$$

How is

$$s \models \forall \Diamond f$$

defined?

Question

Recall that

$$\forall\Diamond f = \forall(\text{true} \; \mathsf{U} \; f)$$

How is

$$s \models \forall\Diamond f$$

defined?

Answer

$$\forall p \in \mathit{Paths}(s) : \exists i \geq 0 : p[i] \models f.$$

Question

Recall that

$$\exists\Box f = \neg\forall(\text{true}\ \mathsf{U}\ \neg f)$$

How is

$$s \models \exists\Box f$$

defined?

Semantics of CTL

Question

Recall that

$$\exists\Box f = \neg\forall(\text{true}\ \mathsf{U}\ \neg f)$$

How is

$$s \models \exists\Box f$$

defined?

Answer

$$\exists p \in \mathit{Paths}(s) : \forall i \geq 0 : p[i] \models f.$$

Semantics of CTL

Question

Recall that

$$\forall\Box f = \neg\exists(\text{true}\ \mathsf{U}\ \neg f)$$

How is

$$s \models \forall\Box f$$

defined?

Semantics of CTL

Question

Recall that

$$\forall\Box f = \neg\exists(\text{true}\ \mathrm{U}\ \neg f)$$

How is

$$s \models \forall\Box f$$

defined?

Answer

$$\forall p \in Paths(s) : \forall i \geq 0 : p[i] \models f.$$

Expressiveness of LTL and CTL

Theorem

The property

$$\forall p \in Paths(TS) : \forall m \geq 0 : \exists p' \in Paths(p[m]) : \exists n \geq 0 : p'[n] \models a$$

cannot be captured by LTL, but is captured by the CTL formula $\forall \Box \exists \Diamond a$.

Expressiveness of LTL and CTL

Theorem

The property

$$\forall p \in Paths(TS) : \exists i \geq 0 : \forall j \geq i : p[j..] \models a$$

cannot be captured by CTL, but is captured by the LTL formula $\Diamond\Box a$.

Model checking CTL

Basic idea

Compute $Sat(f)$ by recursion on the structure of f.

$TS \models f$ iff $I \subseteq Sat(f)$.

Alternative view

Label each state with the subformulas of f that it satisfies.

Model checking CTL

Definition

The *formulas* are defined by

$$f ::= a \mid f \wedge f \mid \neg f \mid \exists \bigcirc f \mid \exists (f \, \mathsf{U} \, f) \mid \forall \bigcirc f \mid \forall (f \, \mathsf{U} \, f)$$

Question

What is $Sat(a)$?

Model checking CTL

Definition

The *formulas* are defined by

$$f ::= a \mid f \wedge f \mid \neg f \mid \exists\bigcirc f \mid \exists(f \, \mathsf{U} \, f) \mid \forall\bigcirc f \mid \forall(f \, \mathsf{U} \, f)$$

Question

What is $Sat(a)$?

Answer

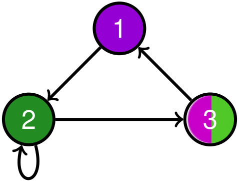
$Sat(a) = \{ s \in S \mid a \in \ell(s) \}$

Alternative view

Label each state s satisfying $a \in \ell(s)$ with a.

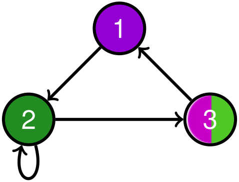
Example

green



Example

green



$$
\begin{array}{lcl}
1 & \mapsto & \emptyset \\
2 & \mapsto & \{\text{green}\} \\
3 & \mapsto & \{\text{green}\}
\end{array}
$$

Model checking CTL

Definition

The *formulas* are defined by

$$f ::= a \mid f \wedge f \mid \neg f \mid \exists \bigcirc f \mid \exists (f \; \mathsf{U} \; f) \mid \forall \bigcirc f \mid \forall (f \; \mathsf{U} \; f)$$

Question

What is $Sat(f_1 \wedge f_2)$?

Model checking CTL

Definition

The *formulas* are defined by

$$f ::= a \mid f \wedge f \mid \neg f \mid \exists \bigcirc f \mid \exists (f \; \mathsf{U} \; f) \mid \forall \bigcirc f \mid \forall (f \; \mathsf{U} \; f)$$

Question

What is $Sat(f_1 \wedge f_2)$?

Answer

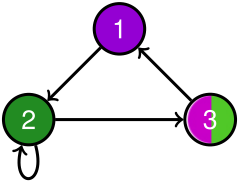
$Sat(f_1 \wedge f_2) = Sat(f_1) \cap Sat(f_2)$

Alternative view

Label states, that are labelled with both f_1 and f_2, also with $f_1 \wedge f_2$.

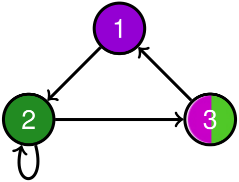
Example

green $\wedge$ purple



Example

$$\text{green} \wedge \text{purple}$$



$$\begin{array}{lcl} 1 & \mapsto & \{\text{purple}\} \\ 2 & \mapsto & \{\text{green}\} \\ 3 & \mapsto & \{\text{green}, \text{purple}, \text{green} \wedge \text{purple}\} \end{array}$$

Model checking CTL

Definition

The *formulas* are defined by

$$f ::= a \mid f \wedge f \mid \neg f \mid \exists\bigcirc f \mid \exists(f \; \mathsf{U} \; f) \mid \forall\bigcirc f \mid \forall(f \; \mathsf{U} \; f)$$

Question

What is $Sat(\neg f)$?

Model checking CTL

Definition

The *formulas* are defined by

$$f ::= a \mid f \wedge f \mid \neg f \mid \exists\bigcirc f \mid \exists(f \,\mathsf{U}\, f) \mid \forall\bigcirc f \mid \forall(f \,\mathsf{U}\, f)$$

Question

What is $Sat(\neg f)$?

Answer

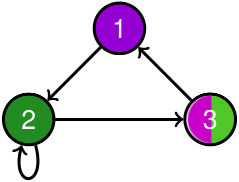
$Sat(\neg f) = S \setminus Sat(f)$

Alternative view

Label each state, that is not labelled with f, with $\neg f$.

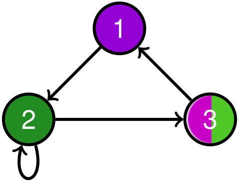
Example

$$\neg(\text{green} \wedge \text{purple})$$



Example

$$\neg(\text{green} \wedge \text{purple})$$



$$\begin{array}{lcl}
1 & \mapsto & \{\text{purple}, \neg(\text{green} \wedge \text{purple})\} \\
2 & \mapsto & \{\text{green}, \neg(\text{green} \wedge \text{purple})\} \\
3 & \mapsto & \{\text{green}, \text{purple}, \text{green} \wedge \text{purple}\}
\end{array}$$

Model checking CTL

Definition

The *formulas* are defined by

$$f ::= a \mid f \wedge f \mid \neg f \mid \exists \bigcirc f \mid \exists (f \; \mathrm{U} \; f) \mid \forall \bigcirc f \mid \forall (f \; \mathrm{U} \; f)$$

Question

What is $Sat(\exists \bigcirc f)$?

Model checking CTL

Definition

The *formulas* are defined by

$$f ::= a \mid f \wedge f \mid \neg f \mid \exists\bigcirc f \mid \exists(f \; \mathsf{U} \; f) \mid \forall\bigcirc f \mid \forall(f \; \mathsf{U} \; f)$$

Question

What is $Sat(\exists\bigcirc f)$?

Answer

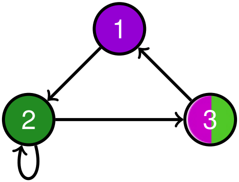
$Sat(\exists\bigcirc f) = \{\, s \in S \mid Post(s) \cap Sat(f) \neq \emptyset \,\}$ where
$Post(s) = \{\, s' \in S \mid s \to s' \,\}$.

Alternative view

Labels those states, that have a direct successor labelled with f, also with $\exists\bigcirc f$.

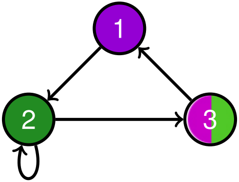
Example

$\exists\bigcirc$green



Example

$$\exists \bigcirc \text{green}$$



$$\begin{array}{lcl} 1 & \mapsto & \{\exists \bigcirc \text{green}\} \\ 2 & \mapsto & \{\text{green}, \exists \bigcirc \text{green}\} \\ 3 & \mapsto & \{\text{green}\} \end{array}$$

Definition

The *formulas* are defined by

$$f ::= a \mid f \wedge f \mid \neg f \mid \exists\bigcirc f \mid \exists(f \; \mathsf{U} \; f) \mid \forall\bigcirc f \mid \forall(f \; \mathsf{U} \; f)$$

Question

What is $Sat(\exists(f_1 \; \mathsf{U} \; f_2))$?

Model checking CTL

$s \in Sat(\exists(f_1 \text{ U } f_2))$

iff $s \models \exists(f_1 \text{ U } f_2)$

iff $s \models f_2 \vee (s \models f_1 \wedge \exists s \to t : t \models \exists(f_1 \text{ U } f_2))$

iff $s \in Sat(f_2) \vee (s \in Sat(f_1) \wedge \exists t \in Post(s) : t \in Sat(\exists(f_1 \text{ U } f_2))$

iff $s \in Sat(f_2) \cup \{ s \in Sat(f_1) \mid Post(s) \cap Sat(\exists(f_1 \text{ U } f_2)) \neq \emptyset\}$

Model checking CTL

$$
\begin{array}{ll}
\multicolumn{2}{l}{s \in Sat(\exists(f_1 \, \mathrm{U} \, f_2))} \\
\text{iff} & s \models \exists(f_1 \, \mathrm{U} \, f_2) \\
\text{iff} & s \models f_2 \vee (s \models f_1 \wedge \exists s \rightarrow t : t \models \exists(f_1 \, \mathrm{U} \, f_2)) \\
\text{iff} & s \in Sat(f_2) \vee (s \in Sat(f_1) \wedge \exists t \in Post(s) : t \in Sat(\exists(f_1 \, \mathrm{U} \, f_2)) \\
\text{iff} & s \in Sat(f_2) \cup \{ s \in Sat(f_1) \mid Post(s) \cap Sat(\exists(f_1 \, \mathrm{U} \, f_2)) \neq \emptyset \}
\end{array}
$$

Proposition

$Sat(\exists(f_1 \, \mathrm{U} \, f_2))$ is the smallest subset T of S such that

$$T = Sat(f_2) \cup \{ s \in Sat(f_1) \mid Post(s) \cap T \neq \emptyset \}.$$

Model checking CTL

$s \in Sat(\exists(f_1 \; \mathrm{U} \; f_2))$

iff $\quad s \models \exists(f_1 \; \mathrm{U} \; f_2)$

iff $\quad s \models f_2 \vee (s \models f_1 \wedge \exists s \to t : t \models \exists(f_1 \; \mathrm{U} \; f_2))$

iff $\quad s \in Sat(f_2) \vee (s \in Sat(f_1) \wedge \exists t \in Post(s) : t \in Sat(\exists(f_1 \; \mathrm{U} \; f_2))$

iff $\quad s \in Sat(f_2) \cup \{\, s \in Sat(f_1) \mid Post(s) \cap Sat(\exists(f_1 \; \mathrm{U} \; f_2)) \neq \emptyset\}$

Proposition

$Sat(\exists(f_1 \; \mathrm{U} \; f_2))$ is the smallest subset T of S such that

$$T = Sat(f_2) \cup \{\, s \in Sat(f_1) \mid Post(s) \cap T \neq \emptyset\}.$$

Question

Does such a smallest subset exist?

Smallest Subset

Definition

The function $F : 2^S \to 2^S$ is defined by

$$F(T) = Sat(f_2) \cup \{\, s \in Sat(f_1) \mid Post(s) \cap T \neq \emptyset \,\}.$$

Definition

A function $G : 2^S \to 2^S$ is monotone if for all $T, U \in 2^S$, if $T \subseteq U$ then $G(T) \subseteq G(U)$.

Smallest Subset

Proposition

F is monotone.

Proof

Let $T, U \in 2^S$. Assume that $T \subseteq U$. Let $s \in F(T)$. It remains to prove that $s \in F(U)$. Then $s \in Sat(f_2)$ or $s \in Sat(f_1)$ and $Post(s) \cap T \neq \emptyset$. We distinguish two cases.

- If $s \in Sat(f_2)$ then $s \in F(U)$.
- If $s \in Sat(f_1)$ and $Post(s) \cap T \neq \emptyset$ then $Post(s) \cap U \neq \emptyset$ since $T \subseteq U$. Hence, $s \in F(U)$.

Smallest Subset

Definition

For each $n \in \mathbb{N}$, the set F_n is defined by

$$F_n = \begin{cases} \emptyset & \text{if } n = 0 \\ F(F_{n-1}) & \text{otherwise} \end{cases}$$

Smallest Subset

Proposition

For all $n \in \mathbb{N}$, $F_n \subseteq F_{n+1}$.

Proof

We prove this by induction on n. In the base case, $n = 0$, we have that

$$F_0 = \emptyset \subseteq F_1 .$$

In the inductive case, we have $n > 1$. By induction, $F_{n-1} \subseteq F_n$. Since F is monotone, we have that

$$F_n = F(F_{n-1}) \subseteq F(F_n) = F_{n+1} .$$

Smallest Subset

Proposition

If S is a finite set. then $F_n = F_{n+1}$ for some $n \in \mathbb{N}$.

Proof

Suppose that S contains m elements. Towards a contradiction, assume that $F_n \neq F_{n+1}$ for all $n \in \mathbb{N}$. Then $F_n \subset F_{n+1}$ for all $n \in \mathbb{N}$. Hence, F_n contains at least n elements. Therefore, F_{m+1} contains more elements than S. This contradicts that $F_{m+1} \subseteq S$.

We denote the F_n with $F_n = F_{n+1}$ by $\mathit{fix}(F)$.

Smallest Subset

Proposition

For all $T \subseteq S$, if $F(T) = T$ then $fix(F) \subseteq T$.

Proof

First, we prove that for all $n \in \mathbb{N}$, $F_n \subseteq T$ by induction on n. In the base case, $n = 0$, we have that

$$F_0 = \emptyset \subseteq T.$$

In the inductive case, we have $n > 1$. By induction, $F_{n-1} \subseteq T$. By induction

$$F_n = F(F_{n-1}) \subseteq F(T) = T.$$

Since $fix(F) = F_n$ for some $n \in \mathbb{N}$, we can conclude that $fix(F) \subseteq T$.

Smallest Subset

Corollary

fix(F) is the smallest T of S such that $F(T) = T$.

$Sat(f)$:
switch (f):

$$
\begin{array}{rcl}
a & : & \textbf{return } \{\, s \in S \mid a \in \ell(s) \,\} \\
f_1 \wedge f_2 & : & \textbf{return } Sat(f_1) \cap Sat(f_2) \\
\neg f & : & \textbf{return } S \setminus Sat(f) \\
\exists \bigcirc f & : & \textbf{return } \{\, s \in S \mid Post(s) \cap Sat(f) \neq \emptyset \,\} \\
\exists(f_1 \,\mathsf{U}\, f_2) & : & T := \emptyset \\
 & & \textbf{while } T \neq F(T) \\
 & & \qquad T := F(T) \\
 & & \textbf{return } T
\end{array}
$$

...

where $F(T) = Sat(f_2) \cup \{\, s \in Sat(f_1) \mid Post(s) \cap T \neq \emptyset \}$.

$Sat(f)$:
switch (f):

$$\begin{array}{lll}
\ldots & & \\
\exists(f_1 \, \mathrm{U} \, f_2) & : & E := Sat(f_2) \\
& & T := E \\
& & \textbf{while } E \neq \emptyset \\
& & \quad \textbf{let } t \in E \\
& & \quad E := E \setminus \{t\} \\
& & \quad \textbf{for all } s \in Pre(t) \\
& & \quad\quad \textbf{if } s \in Sat(f) \setminus T \\
& & \quad\quad\quad E := E \cup \{s\} \\
& & \quad\quad\quad T := T \cup \{s\} \\
& & \textbf{return } T \\
\ldots & &
\end{array}$$

where $Pre(t) = \{ s \in S \mid s \to t \}$.

Model checking CTL

Definition

The *formulas* are defined by

$$f ::= a \mid f \wedge f \mid \neg f \mid \exists \bigcirc f \mid \exists (f \mathbin{\mathrm{U}} f) \mid \forall \bigcirc f \mid \forall (f \mathbin{\mathrm{U}} f)$$

Question

What is $Sat(\forall \bigcirc f)$?

Definition

The *formulas* are defined by

$$f ::= a \mid f \wedge f \mid \neg f \mid \exists \bigcirc f \mid \exists (f \ \mathsf{U}\ f) \mid \forall \bigcirc f \mid \forall (f \ \mathsf{U}\ f)$$

Question

What is $Sat(\forall \bigcirc f)$?

Answer

$Sat(\forall \bigcirc f) = \{\, s \in S \mid Post(s) \subseteq Sat(f)\,\}$.

Definition

The *formulas* are defined by

$$f ::= a \mid f \wedge f \mid \neg f \mid \exists \bigcirc f \mid \exists (f \, \mathsf{U} \, f) \mid \forall \bigcirc f \mid \forall (f \, \mathsf{U} \, f)$$

Question

What is $Sat(\forall (f_1 \, \mathsf{U} \, f_2))$?

Model checking CTL

$$
\begin{array}{ll}
\multicolumn{2}{l}{s \in Sat(\forall(f_1 \mathbin{\mathsf{U}} f_2))} \\
\text{iff} & s \models \forall(f_1 \mathbin{\mathsf{U}} f_2) \\
\text{iff} & s \models f_2 \vee (s \models f_1 \wedge \forall s \to t : t \models \forall(f_1 \mathbin{\mathsf{U}} f_2)) \\
\text{iff} & s \in Sat(f_2) \vee (s \in Sat(f_1) \wedge \forall t \in Post(s) : t \in Sat(\forall(f_1 \mathbin{\mathsf{U}} f_2)) \\
\text{iff} & s \in Sat(f_2) \cup \{\, s \in Sat(f_1) \mid Post(s) \subseteq Sat(\forall(f_1 \mathbin{\mathsf{U}} f_2))\}
\end{array}
$$

Model checking CTL

$$
\begin{aligned}
& s \in Sat(\forall(f_1 \mathrm{~U~} f_2)) \\
& \quad \text{iff} \quad s \models \forall(f_1 \mathrm{~U~} f_2) \\
& \quad \text{iff} \quad s \models f_2 \vee (s \models f_1 \wedge \forall s \rightarrow t : t \models \forall(f_1 \mathrm{~U~} f_2)) \\
& \quad \text{iff} \quad s \in Sat(f_2) \vee (s \in Sat(f_1) \wedge \forall t \in Post(s) : t \in Sat(\forall(f_1 \mathrm{~U~} f_2)) \\
& \quad \text{iff} \quad s \in Sat(f_2) \cup \{\, s \in Sat(f_1) \mid Post(s) \subseteq Sat(\forall(f_1 \mathrm{~U~} f_2))\}
\end{aligned}
$$

Proposition

$Sat(\forall(f_1 \mathrm{~U~} f_2))$ is the smallest subset T of S such that

$$T = Sat(f_2) \cup \{\, s \in Sat(f_1) \mid Post(s) \subseteq T\}.$$

Model checking CTL

$s \in Sat(\forall(f_1 \mathrel{\mathsf{U}} f_2))$

iff $s \models \forall(f_1 \mathrel{\mathsf{U}} f_2)$

iff $s \models f_2 \vee (s \models f_1 \wedge \forall s \to t : t \models \forall(f_1 \mathrel{\mathsf{U}} f_2))$

iff $s \in Sat(f_2) \vee (s \in Sat(f_1) \wedge \forall t \in Post(s) : t \in Sat(\forall(f_1 \mathrel{\mathsf{U}} f_2))$

iff $s \in Sat(f_2) \cup \{ s \in Sat(f_1) \mid Post(s) \subseteq Sat(\forall(f_1 \mathrel{\mathsf{U}} f_2))\}$

Proposition

$Sat(\forall(f_1 \mathrel{\mathsf{U}} f_2))$ is the smallest subset T of S such that

$$T = Sat(f_2) \cup \{ s \in Sat(f_1) \mid Post(s) \subseteq T\}.$$

Question

Does such a smallest subset exist?

Size of a CTL formula

$$
\begin{array}{rcl}
|a| & = & 1 \\
|f_1 \wedge f_2| & = & 1 + |f_1| + |f_2| \\
|\neg f| & = & 1 + |f| \\
|\exists\bigcirc f| & = & 1 + |f| \\
|\forall\bigcirc f| & = & 1 + |f| \\
|\exists\bigcirc(f_1 \; \mathsf{U} \; f_2)| & = & 1 + |f_1| + |f_2| \\
|\forall\bigcirc(f_1 \; \mathsf{U} \; f_2)| & = & 1 + |f_1| + |f_2|
\end{array}
$$

Time Complexity of CTL Model Checking

By improving the model checking algorithm (see, for example the textbook of Baier and Katoen for details), we obtain

Theorem

For a transition system TS, with N states and K transitions, and a CTL formula f, the model checking problem $TS \models f$ can be decided in time $\mathcal{O}((N + K) \cdot |f|)$.

Time Complexity of CTL Model Checking

By improving the model checking algorithm (see, for example the textbook of Baier and Katoen for details), we obtain

Theorem

For a transition system TS, with N states and K transitions, and a CTL formula f, the model checking problem $TS \models f$ can be decided in time $\mathcal{O}((N + K) \cdot |f|)$.

Theorem

For a transition system TS, with N states and K transitions, and a LTL formula g, the model checking problem $TS \models g$ can be decided in time $\mathcal{O}((N + K) \cdot 2^{|g|})$.

Time Complexity of CTL Model Checking

By improving the model checking algorithm (see, for example the textbook of Baier and Katoen for details), we obtain

Theorem

For a transition system TS, with N states and K transitions, and a CTL formula f, the model checking problem $TS \models f$ can be decided in time $\mathcal{O}((N + K) \cdot |f|)$.

Theorem

For a transition system TS, with N states and K transitions, and a LTL formula g, the model checking problem $TS \models g$ can be decided in time $\mathcal{O}((N + K) \cdot 2^{|g|})$.

Theorem

If P $\neq$ NP then there exist LTL formulas g_n whose size is a polynomial in n, for which equivalent CTL formulas exist, but not of size polynomial in n.