Introduction to Linear Temporal Logic

Youhei Okazaki
FL11-16-1
25th, October, 2011

Outline

Introduction: Atomic Formula

Predicate Logic

term: x
predicate: p
atomic formula: $p(x) \in \{T, F\}$

Ex.
A swallow (x) is a bird (p).
A cock (x) is a bird (p).
A sparrow (x) can fly (p).

each atomic formula value $\{T, F\}$ is decided by selection of x and p

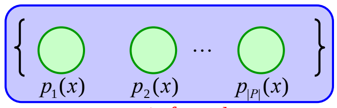
a proposition is denoted as an atomic formula $p(x)$

predicate index: i $p_i(x)$

Introduction: Propositional Logic

Predicates

about term x P



set of predicate: P
cardinality of P: $|P|$
predicate: $p_i, i \in \{1, 2, \ldots, |P|\}$
set of atomic formula: $P(x) := \{p_i(x) \mid p_i \in P\}$ atomic formulae

each atomic formula is binary valued variable (ex. true or false)
$p_i(x) \in \{T, F\}$

Logical Connective

negation: $\neg$ ($\neg p_1(x) = T$ holds iff $p_1(x) = F$ hold)
conjunction("and"): $\wedge$ ($p_1 \wedge p_2(x) = T$ holds iff both $p_1(x) = T$ and $p_2(x) = T$ hold)
disjunction("or"): $\vee$ $(p_1 \vee p_2 := \neg(\neg p_1 \wedge \neg p_2))$
implication: $\rightarrow$ $(p_1 \rightarrow p_2 := \neg p_1 \vee p_2)$

Introduction: Propositional Logic

Propositional Logic Formula

Propositional logic formula over set P of atomic proposition are formed using logical connective
propositional logic formula: φ

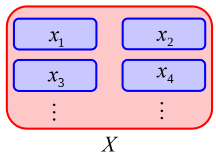
value of φ in term x: $\varphi(x) \in \{T, F\}$

Truth Table formulae in term x_1, x_2, x_3, x_4

x	$p_1(x)$	$p_2(x)$	$\neg p_1(x)$	$p_1 \wedge p_2(x)$	$p_1 \vee p_2(x)$	$p_1 \rightarrow p_2(x)$
x_1	T	T	F	T	T	T
x_2	T	F	F	F	T	F
x_3	F	T	T	F	T	T
x_4	F	F	T	F	F	T

Introduction: Terms

Terms



set of terms: X
term: $x_m \in X$
term index: $m \in \{1, 2, \ldots, |X|\}$

Proposition on terms

Modal Operator

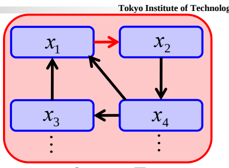
φ is formula in $x_m \in X$
may: $\Diamond$ $\Diamond\varphi(X)$ holds iff $\varphi(x_m) = T, \exists x_m \in X$
must: $\Box$ $\Box\varphi(X)$ holds iff $\varphi(x_m) = T, \forall x_m \in X$

Introduction: Frame

Accessibility

set of accessible relations: $R \subseteq X \times X$

Ex. $(x_1, x_2) \in R$ (x_2 is accessible for x_1)



frame: F

Frame

Frame: $F := (X, R)$

$F(x) := \{x_m \mid (x, x_m) \in R\} \subseteq X$

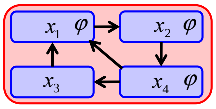
Modal Operator

φ is formula in $x_m \in X$

may: ◊ $\Diamond\varphi(x) := \Diamond\varphi(F(x))$ holds iff there exists $x_m \in F(x)$ and in this condition $\varphi(x_m) = \mathrm{T}$

must: □ $\Box\varphi(x) := \Box\varphi(F(x))$ holds iff $\varphi(x_m) = \mathrm{T}, \forall x_m \in F(x)$

Ex.



$\Box\varphi(x_1) = \mathrm{T}$

$\Diamond\varphi(x_3) = \mathrm{T}$

$\Box\varphi(x_4) = \mathrm{F}$

Introduction: Temporal Logic

Trace

set of atomic formulae satisfied in the term: $p(x_m) := \{p_i \in P \mid p_i(x_m) = T, i \in \{0,1,\ldots,|P|\}\}$

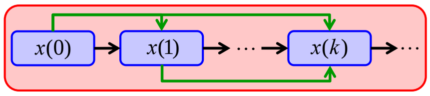
trace σ $p(x(0)) \rightarrow p(x(1)) \rightarrow \cdots \rightarrow p(x(k)) \rightarrow p(x(k+1)) \rightarrow \cdots$

k_{th} term $x(k) \in X$

Temporal Axiom about R

- irreflexive: in any $x(k)$, $(x(k), x(k)) \notin R$
- transitive: if $(x(k), x(k')) \in R$ and $(x(k'), x(k'')) \in R$ then $(x(k), x(k'')) \in R$
- connected: between any moments k, k', $(x(k), x(k')) \in R$ or $(x(k'), x(k)) \in R$ or k, k'

trace + axiom ⇒ frame



Introduction: Linear Temporal Logic

Semantics

$\sigma = p(x(0))p(x(1))\ldots p(x(k))\ldots$

$\sigma[j\ldots] = p(x(j))p(x(j+1))\ldots p(x(k))\ldots, j \geq 0$

φ is formula in $x_m \in X$

$\varphi(\sigma)$ holds iff $\varphi(x(0)) = \mathrm{T}$

LTL Operator

eventually: ◊ $\Diamond\varphi(\sigma)$ holds iff $\varphi(\sigma[j\ldots]) = \mathrm{T}, \exists j \geq 0$

always: □ $\Box\varphi(\sigma)$ holds iff $\varphi(\sigma[j\ldots]) = \mathrm{T}, \forall j \geq 0$

next: ○ $\bigcirc\varphi(\sigma)$ holds iff $\varphi(\sigma[1\ldots]) = \mathrm{T}$

until: *Until* $\varphi_1 \, Until \, \varphi_2(\sigma)$ holds iff $\varphi_2(\sigma[j_2\ldots]) = \mathrm{T}, \exists j_2 \geq 0$ and $\varphi_1(\sigma[j_1\ldots]) = \mathrm{T}$, for all $0 \leq j_1 < j_2$

LTL Formula

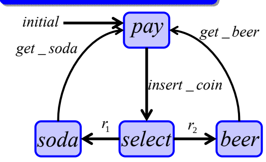
LTL formula Φ $\boxed{\varphi \ \neg \ \wedge \ \vee \ \rightarrow \ \Diamond \ \Box \ \bigcirc \ Until}$ $\Phi(\sigma) \in \{T, F\}$

Outline

- Introduction to Temporal Logic
- **Verification and Control**
- N. Ozay et al.
- Y. Chen et al.
- Summary

Plant Model: Example

Vending Machine



initial state $X^{in} := \{pay\}$

state $X := \{pay, select, soda, beer\}$

input $U := \{insert\_coin, get\_soda, get\_beer, r_1, r_2\}$

transition relation $T := \left\{ pay \xrightarrow{insert\_coin} select, \ beer \xrightarrow{get\_beer} pay, \ \vdots \right\}$

Desired Property

"The Vending Machine only delivered a drink after providing a coin"

predicate $P := \{paid, drink\}$

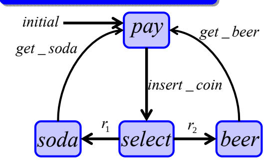
labeling $L(pay) := \phi, L(select) := \{paid\}, L(soda) := \{paid, drink\}, L(beer) := \{paid, drink\}$

$\Phi := \Box\{\neg drink\} Until \{paid\}$

goal $\Phi(\Gamma) = \mathrm{T}$

Execution: Example

Execution



system model $\Gamma : (X, U, T, X^{in}, P, L)$

execution

path in Γ

$$\left\{ \begin{array}{l} pay \xrightarrow{coin} select \xrightarrow{r_1} soda \xrightarrow{get} pay \xrightarrow{coin} select \ldots \\ select \xrightarrow{r_1} soda \xrightarrow{get} pay \xrightarrow{coin} select \xrightarrow{r_2} beer \ldots \\ pay \xrightarrow{coin} select \xrightarrow{r_2} beer \xrightarrow{get} pay \xrightarrow{coin} select \\ \vdots \end{array} \right.$$

path $r_\Gamma = pay \; select \; soda \; pay \; select \ldots$

trace $trace(r_\Gamma) = L(pay) \; L(select) \; L(soda) \; L(pay) \; L(select) \ldots$

Traces: Example

Traces

Paths

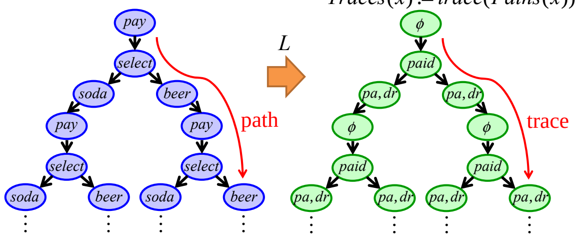
$Paths(\Gamma) := \{Paths(x), x \in X^{in}\}$

$Paths(x)$: set of all paths start from x

Traces

$Traces(\Gamma) := \bigcup_{x \in X^{in}} Traces(x)$

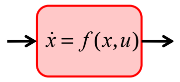
$Traces(x) := trace(Paths(x))$



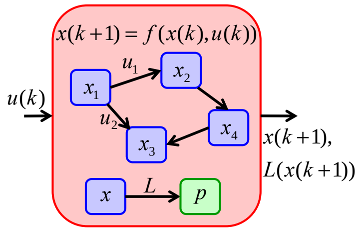
Semantics of LTL over Paths $\Phi(\Gamma) = \mathrm{T}$ iff Φ is satisfied $\forall r_\Gamma \in Paths(\Gamma)$

Plant Model

Plant Model



Simplified Plant Model



set of states: X

set of inputs: U

transition relation: $T \subseteq X \times U \times X$

set of initial states: $X^{in} \subseteq X$

set of predicates: P

labeling function: $L : X \to 2^P$

state: $x \in X$ input: $u \in U$

set of satisfied atomic formulae: $p \subseteq P$

transition function $f := \{x(k+1) \mid (x(k), u(k), x(k+1)) \in T\}$

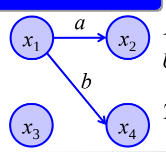
Transition System

Transition System system model is simplified to discrete valued valuable model

- X : set of states
- U : set of inputs
- $T \subseteq X \times U \times X$: transition relation
- $X^{in} \subseteq X$: set of initial states
- P : set of predicates
- $L : X \to 2^P$: labeling function

$\Gamma : (X, U, T, X^{in}, P, L)$ transition system

Transition Relation



$X := \{x_1, x_2, x_3, x_4\}$

$U := \{a, b\}$

$T := \left\{ \begin{array}{l} (x_1, a, x_2), \\ (x_1, b, x_4) \end{array} \right\}$

Atomic Formulae

$P := \{p_1, p_2\}$

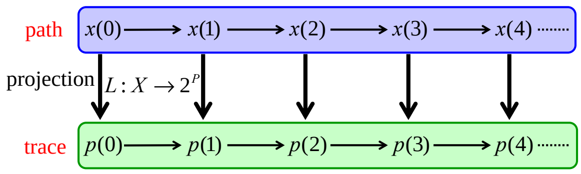
$L : X \to 2^P$:

x	x_1	x_2	x_3	x_4
	$L(x)$ ↓			
p_1	T	F	T	F
p_2	T	T	F	F

Path and Trace

Path and Trace

$x(0) \in X^{in}$



satisfied predicates $p(k) = L(x(k))$

Traces

Traces

Paths

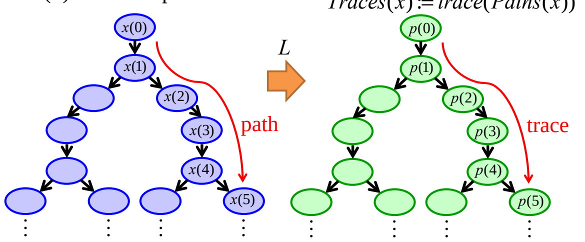
$Paths(\Gamma) := \{Paths(x), x \in X^{in}\}$

$Paths(x)$: set of all paths start from x

Traces

$Traces(\Gamma) := \bigcup_{x \in X^{in}} Traces(x)$

$Traces(x) := trace(Paths(x))$



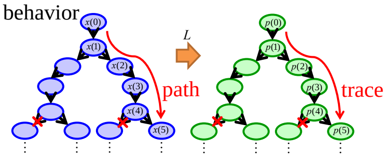
Semantics of LTL over Paths $\Phi(\Gamma) = \mathrm{T}$ iff Φ is satisfied $\forall r_\Gamma \in Paths(\Gamma)$

Problem

Given

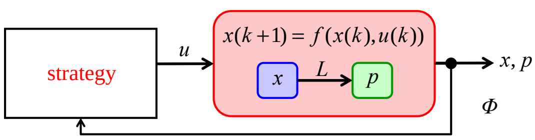
LTL formula Φ: desired behavior

system model Γ: plant



Goal

derive strategy to determine u satisfied Φ from an algorithmic framework



Outline

- Introduction to Temporal Logic
- Verification and Control
- **N. Ozay et al.**
- Y. Chen et al.
- Summary

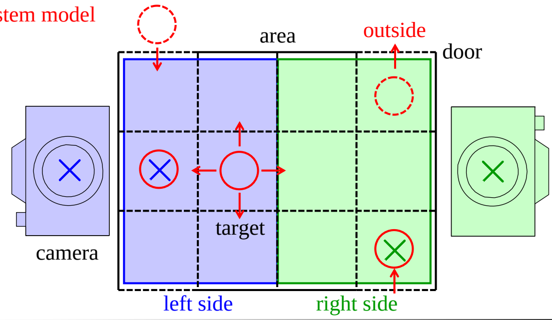
N. Ozay et al. [1]

Camera Network

desired behavior

"All Target should be zoomed before go out from the area"

system model



N. Ozay et al. [1]

Rule

1. At each time step a target either remains in the previous cell or moves to a neighboring cell
2. Every target always eventually exits the area
3. Every target can exit and enter through designated doors
4. A target remains in the area at least K time step
5. There could be at most N targets on each side of the area
6. The camera zooms into single cell
7. Once a target is zoomed-in, image of the target is taken
8. After a target goes out from the area, the image is deleted

Goal

"All Target should be zoomed before go out from the area"

N. Ozay et al. [1]

State $\Gamma : (X, U, T, X^{in}, P, L)$

Area

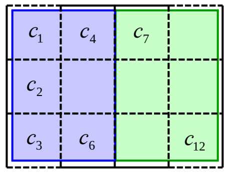
Cell Position $C := \{c_1, c_2, \ldots, c_{12}\}$

Left Camera Area

$C_l := \{c_1, c_2, \ldots, c_6\}$

Right Camera Area

$C_r := \{c_7, c_8, \ldots, c_{12}\}$



Target and Camera

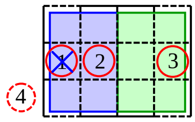
Target $i \in \{1, 2, \ldots, 2N\}$

Target i 's position at time k $a_i(k) \in C \cup \{c_0\}$

Number of time step target i remains $n_i(k) \in \{0, 1, 2, 3\}$

Camera $j \in \{l, r\}$

Camera zoom position at time k $z_j(k) \in C_j$



Ex. at time k

$a_1(k) = c_2$

$a_4(k) = c_0$

$z_l(k) = c_2$

N. Ozay et al. [1]

State $\Gamma : (X, U, T, X^{in}, P, L)$

Zoomed Image

Taken image $isZoomed_i(k) := \{T, F\}$

State

$$X := \prod_i ((C \cup \{c_0\}) \times \{0,1,2,3\} \times \{T, F\}) \times \prod_j \{C_j\}$$

$x(k) \in X$

$$x(k) = (a_1(k), a_2(k), \ldots, a_{2N}(k), n_1(k), n_2(k), \ldots, n_{2N}(k), isZoomed_1(k), \ldots, isZoomed_{2N}(k), z_l(k), z_r(k))$$

$$X_e := \prod_i ((C \cup \{c_0\}) \times \{0,1,2,3\} \times \{T, F\}), X_p := \prod_j \{C_j\}$$

$C := \{c_1, c_2, \ldots, c_{12}\}$

$i \in \{1, 2, \ldots, 2N\}$

$a_i(k) \in C \cup \{c_0\}$

$n_i(k) \in \{0, 1, 2, 3\}$

$j \in \{l, r\}$

$C_l := \{c_1, c_2, \ldots, c_6\}$

$C_r := \{c_7, c_8, \ldots, c_{12}\}$

$z_j(k) \in C_j$

N. Ozay et al. [1]

Rule $\Gamma : (X, U, T, X^{in}, P, L)$

1. At each time step a target either remains in the previous cell or moves to a neighboring cell and rule 3, 5

$$\Psi_{x,i} := \wedge_{m \in \{0,1,\ldots,12\}} \Box((a_i(k) = c_m) \rightarrow \bigcirc(\vee_{l \in L_m} (a_i(k) = c_l)))$$

L_m : the set of the cells target is allowed to move from c_m

2. Every target always eventually leaves the area

$$\Psi_{l,i} := \Box \Diamond a_i(k) = c_0$$

4. A target remains in the area at least K time step

$$\Psi_{r,i} := \Box(((a_i(k) \neq c_0) \wedge (n_i(k) < K)) \rightarrow \bigcirc (a_i(k) \neq c_0))$$

$$\Psi_{k,i} := \Box[((a_i(k) = c_0) \rightarrow (n_i(k) = 0)) \wedge ((\bigcirc a_i(k) \neq c_0) \rightarrow (\bigcirc(n_i(k)) = \min(n_i(k) + 1, 3)))]$$

6. The camera zoom into single cell

$$\Psi_{c,j} := \Box \vee_{l \in C_j} (z_j(k) = c_l) \quad \text{for } j \in \{l, r\}$$

Rule

7. Once a target is zoomed-in image of the target is taken, and rule 8

$o_{1,i} := ((a_i(k)=z_l(k)) \vee (a_i(k)=z_r(k))) \to (\bigcirc isZoomed_i(k))$

$o_{2,i} := (isZoomed_i(k) \wedge (a_i(k)\neq c_0) \wedge \bigcirc(a_i(k)\neq c_0))$
$\to (\bigcirc isZoomed_i(k))$

$o_{3,i} := ((a_i(k)\neq z_l(k)) \wedge (a_i(k)\neq z_r(k)) \wedge \neg isZoomed_i(k))$
$\to (\bigcirc \neg isZoomed_i(k))$

$o_{4,i} := (a_i(k)=c_0) \to (\bigcirc \neg isZoomed_i(k))$

$\Psi_{o,i} := \wedge_{n\in\{1,\dots,4\}} \Box o_{n,i}$

N. Ozay et al. [1]
Tokyo Institute of Technology

Initial State $\Gamma : (X, U, T, X^{in}, P, L)$

$\Psi_{init,i} := (a_i(k)=c_0) \wedge (n_i(k)=0) \wedge \neg isZoomed_i(k)$

Goal "All Target should be zoomed before go out from the area"

Escape

$escape_i := (a_i(k)\neq c_0) \wedge (\neg isZoomed_i(k)) \wedge \bigcirc(a_i(k)=c_0) \wedge$
$\bigcirc(\neg isZoomed_i(k))$

Goal $\Psi_g := \Box(\wedge_{i\in\{1,2,\dots,2N\}} \neg escape_i(k))$

Formula $\Gamma : (X, U, T, X^{in}, P, L)$

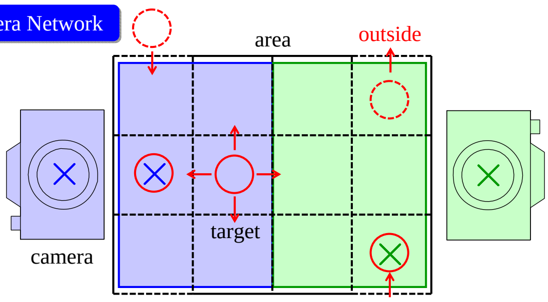
$\Phi_e := \wedge_{i\in\{1,2,\dots,2N\}} (\Psi_{x,i} \wedge \Psi_{l,i} \wedge \Psi_{r,i} \wedge \Psi_{k,i} \wedge \Psi_{o,i} \wedge \Psi_{init,i})$: environment assumption

$\Phi_s := \Psi_g \wedge \Psi_{c,l} \wedge \Psi_{c,r}$: desired system behavior

Desired Property $\Phi := \Phi_e \to \Phi_s$ (Φ_s is satisfied in Γ)

N. Ozay et al. [1]
Tokyo Institute of Technology

Camera Network



controllable input: $z_j(k)\in C_j \quad j\in\{l,r\}$ X_p

not controllable: $a_i(k), n_i(k), isZoomed_i(k) \quad i\in\{1,2,\dots,2N\}$ X_e under rule

N. Ozay et al. [1]
Tokyo Institute of Technology

Camera Network Γ

plant(controllable): camera, environment(not controllable): target

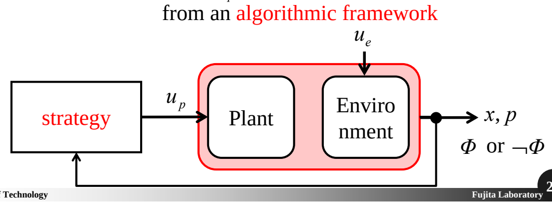
Desired Property

"All Target should be zoomed before go out from the area"

formula Φ: desired behavior

Goal

derive strategy to determine u_p satisfied Φ under any u_e from an algorithmic framework



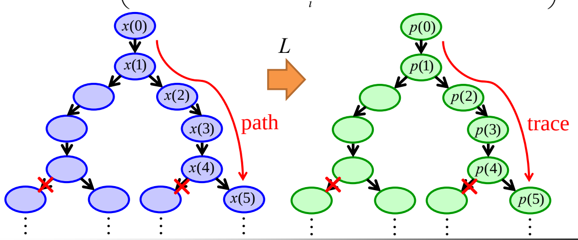
N. Ozay et al. [1]
Tokyo Institute of Technology

Centralized control strategy synthesis

formula Φ iff $(\Phi_e \to \Phi_s)$ Φ_e: environment assumption
Φ_s: desired system behavior

derive strategy, satisfied formula

$$u_p(t) = f\left(x(0)x(1)\dots x(k-1), \prod_i (a_i(k) n_i(k) isZoomed_i(k))\right) \in X_p$$



N. Ozay et al. [1]
Tokyo Institute of Technology

Distributed State

State

$X_j := \prod_i ((C_j \cup \{c_0\}) \times \{0,1,2,3\} \times \{T, F\}) \times \{C_j\}$

$x_j(k) \in X_j$

$x_j(k) = (a_{j1}(k), a_{j2}(k), \dots, a_{j2N}(k),$
$n_1(k), n_2(k), \dots, n_{2N}(k),$
$isZoomed_1(k), \dots, isZoomed_{2N}(k),$
$z_j(k))$

$i\in\{1,2,\dots,2N\}$
$a_{ji}(k)\in C_j \cup \{c_0\}$
$n_i(k)\in\{0,1,2,3\}$
$j\in\{l,r\}$
$C_l := \{c_1, c_2, \dots, c_6\}$
$C_r := \{c_7, c_8, \dots, c_{12}\}$
$z_j(k)\in C_j$

N. Ozay et al. [1]

Centralized control strategy synthesis

formula Φ iff $(\Phi_e \to \Phi_s)$ — Φ_e: environment assumption; Φ_s: desired system behavior

derive strategy, satisfied formula

$$u_p(t) = f\left(x(0)x(1)\ldots x(k-1), \prod_i (a_i(k)n_i(k)isZoomed_i(k)) \right) \in X_p$$

Distributed control strategy synthesis

global property $\Phi = (\Phi_e \to \Phi_s)$

derive distributed strategy, satisfied global property

$$u_{pl}(t) = f(x_l(0)x_l(1)\ldots x_l(k-1), \prod_{i\in left}(a_i(k)n_i(k)isZoomed_i(k))) \in X_{pl}$$

$$u_{pr}(t) = f(x_r(0)x_r(1)\ldots x_r(k-1), \prod_{i\in right}(a_i(k)n_i(k)isZoomed_i(k))) \in X_{pr}$$

$u_p(t) = (u_{pr}(t), u_{pl}(t))$ property $\Phi = (\Phi_e \to \Phi_s)$

N. Ozay et al. [1]

Proposition 1.[1]

Let Φ_e, Φ_{el}, Φ_{er}, Φ_s, Φ_{sl}, and Φ_{sr} be LTL formula that contain variables of only from the respective sets of not controllable varibales X_e, X_{el}, X_{er} and system variables X, X_l, X_r. Let X_p, X_{pl}, X_{pr} be the sets of all controllable variables in X, X_l, X_r that satisfy

$$X_{pl} \cup X_{pr} = X_p, X_{pl} \cap X_{pr} = \phi$$

If the conditions:

- any execution of the environment that satisfies Φ_e, also satisfies $\Phi_{el} \wedge \Phi_{er}$
- any executions of the system that satisfies $\Phi_{sl} \wedge \Phi_{sr}$, also satisfies Φ_s
- and, there exist two control protocols that make the local specifications $(\Phi_{el} \to \Phi_{sl})$ and $(\Phi_{er} \to \Phi_{sr})$ true,

hold, then implementing these two control protocols together would lead to a system where the global specification $\Phi \equiv (\Phi_e \to \Phi_s)$ is met

N. Ozay et al. [1]

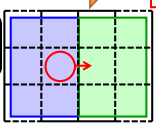
Distributed synthesis

distributed property ⇒ distributed synthesis

$(\Psi'_r \wedge \Phi_{el}) \to (\Phi_{sl} \wedge \Psi_l)$ ⇒ $u_{pl}(t)$

$(\Psi'_r \wedge \Phi_{er}) \to (\Phi_{sr} \wedge \psi_r)$ ⇒ $u_{pr}(t)$

tautology $\left(\begin{array}{l} \Psi_l \to \Psi'_l \\ \Psi_r \to \Psi'_r \end{array} \right)$



using this input, desired behavior is satisfied

Summary

In this way, derive distributed strategy to determine u satisfied Φ from a distributed algorithmic framework

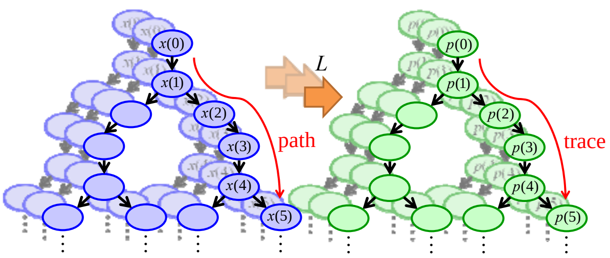
Outline

Y. Chen et al.[2]

cc-strategy $\Gamma:(X,U,T,X^{in},P,L)$ $\{P_i \subseteq P, i\in I\}$

Model each agent as a transition system

$\Gamma_i:(X_i,U_i,T_i,X_i^{in},P_i,L_i)$ $i\in I$



Y. Chen et al.[2]

Global behavior

given a set of cc-strategies $\{r_{\Gamma_i} = x_i(0)x_i(1)x_i(2)x_i(3)\cdots, i\in I\}$,
we denote $l_{team}(\{r_{\Gamma_i}, i\in I\}) := \|_{i\in I} \{w_i\}$
as the set of all possible sequences of propositions satisfied by the

Definition. satisfying set of cc-strategies [2]

A set of cc-strategies $\{r_{\Gamma_i}, i\in I\}$ satisfies a specification given as an LTL formula φ if and only if $l_{team} \neq \phi$ and $l_{team} \subseteq l(B_\varphi)$

implementability distributability

Problem

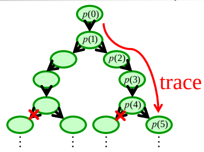
Given a team of agents represented by $\Gamma_i, i\in I$,
a global specification φ in the form of an LTL formula over P and
a distribution $\{P_i \subseteq P, i\in I\}$,

find a satisfying set of individual cc-strategies $\{r_{\Gamma_i}, i\in I\}$

Y. Chen et al.[2]

Synthesis

the global specification φ over P



conversion

$B_\varphi = (Q, Q^{in}, P, \delta, F)$

which accepts exactly the language satisfying φ

We need to find a local trace w_i for each agent i such that

1.all possible sequences of propositions satisfied by the team while each agent executes its local trace satisfy the global specification

2.each local trace can be implemented by the corresponding agent w_i

checking distributability B_φ checking implementability

decide $r_i = x_i(0), x_i(1), \ldots$ of Γ_i

Y. Chen et al.[2]

Synthesis

given the converted global specification $l(B_\varphi)$
and the distribution $\{P_i \subseteq P, i \in I\}$

Proposition 1. [2]

Given a distribution $\{P_i \subseteq P, i \in I\}$ and a trace $w \in P^\infty$, we have

$$[w] = \|_{i \in I} \{w|_{P_i}\}$$

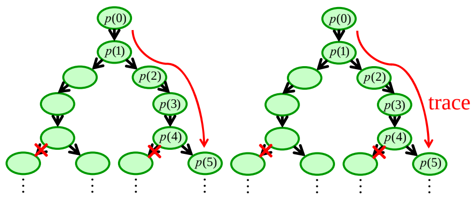
A trace-closed language is sufficient to find a set of local traces satisfying the distributability.

Proposition 2. [2]

Given a language $l \subset P^\infty$ and a distribution $\{P_i \subseteq P, i \in I\}$
if l is trace-closed language and $w \in l$, then $\|_i \{w|_{P_i}\} \subseteq l$

Y. Chen et al.[2]

Trace-equivalent



Y. Chen et al.[2]

Synthesis

Proposition 2. [2]

Given a language $l \subset P^\infty$ and a distribution $\{P_i \subseteq P, i \in I\}$
if l is trace-closed language and $w \in l$, then $\|_i \{w|_{P_i}\} \subseteq l$

Check whether $l(B_\varphi)$ is trace-closed

if the answer is positive, by Prop. 2

an arbitrary trace from $l(B_\varphi)$ can be used to generate suitable set of local traces by projecting trace onto P_i

Y. Chen et al.[2]

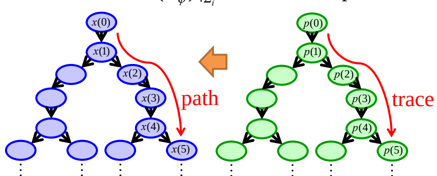
Checking implementability

Proposition 3. [2]

If a set of cc-strategies $\{r_{\Gamma_i}, i \in I\}$ is a solution to problem, then the corresponding local traces r_{Γ_i} are included in $l(B_\varphi)|_{\Sigma_i}, \forall i \in I$

Given the agent model Γ_i some of the local traces might not be feasible for the agent

capture all the traces of $l(B_\varphi)|_{\Sigma_i}$ that can be implemented by the agent



Y. Chen et al.[2]

Theorem 1. [2]

If $l(B_\varphi)$ is trace-closed, the set of cc-strategies $\{r_{\Gamma_i}, i \in I\}$ satisfies $\|_{i \in I} \{w_i\} \neq \phi$ and $\|_{i \in I} \{w_i\} \subseteq l(B_\varphi)$, where w_i is the corresponding trace of Γ_i generated by r_{Γ_i}

Summary

In this way, derive distributed strategy satisfied ϕ.
In addition, from implementability there is input u to follow this strategy, so ϕ is satisfied by input u.
Because this algorithmic framework consider all agents, it is not distributed framework.

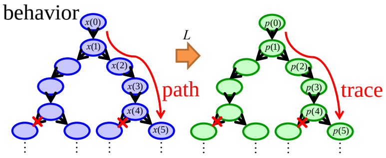
Outline

- Introduction to Temporal Logic
- Verification and Control
- N. Ozay et al.
- Y. Chen et al.
- **Summary**

Summary

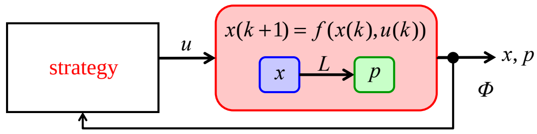
Given

LTL formula Φ: desired behavior
system model Γ: plant



Goal

derive strategy to determine u satisfied Φ
from an algorithmic framework



Reference

Reference

[1] N. Ozay, U. Topcu, T. Wongpiromsarn and R. M. Murray, “Distributed Synthesis of Control Protocols for Smart Camera Networks,” *Proc. of 2011 IEEE/ACM International Conference on Cyber-Physical Systems*, pp. 45 - 54, 2011.
[2] Y. Chen, X. C. Ding and C. Belta, “Synthesis of Distributed Control and Communication Schemes from Global LTL Specifications,” *Proc. of the 50th IEEE Conference on Decision and Control*, 2011. (submitted)
[3] N. Ozay, U. Topcu and R. M. Murray, “Distributed Power Allocation for Vehicle Management Systems,” *Proc. of the 50th IEEE Conference on Decision and Control*, 2011. (submitted)
[4] C. Baier and J. Katoen, *Principles of model checking*, The MIT Press, 2008.
[5] 東条 敏, 言語･知識･信念の論理, オーム社, 2006