基于 GR(1) 规约的控制程序综合

形式化方法课程

2021 年 7 月

程序综合

- 评价
  - ☞ 程序综合 (program synthesis)是指使用指定的编程语言自动生成符合程序规约的技术，……，一直是计算机科学的核心问题和挑战，被认为是计算机科学的圣杯。

    ——*Wang J, Zhan NJ, Feng XY, Liu ZM. Overview of Formal Methods. Journal of Software, 2019, 30(1): 33-61(in Chinese).*
  - ☞ One of the most ambitious and challenging problems in computer science is the automatic synthesis of programs and (digital) designs from logical specifications.

    ——*Bloem R, Jobstmann B, Piterman N, et al. Synthesis of Reactive(1) designs[J]. Journal of Computer & System Sciences, 2012, 78(3):911-938.*

程序综合

- 发展
  - ☞ 基于转换的程序综合 (transformation-based synthesis)，将高层程序规约转换为较低层程序规约，最终生成期望的程序代码。
  - ☞ 规约间或规约与程序间的转换要么归结为树自动机接受语言判空问题，要么归结为两个玩家博弈取胜的策略问题。但这种方法能够自动生成的代码非常有限，很长一段时间没有发展。
  - ☞ 近些年，随着 Pnueli 提出由时序逻辑公式自动综合反应式系统控制程序的成功，有复苏迹象。
- 其他方向
  - ☞ 基于演绎推理的演绎综合(deductive synthesis) 方法：首先，使用定理证明器构造用户提供的程序规约的一个证明；然后，基于 Curry-Howard 同构关系，使用该证明来生成相应的程序代码。
  - ☞ 归纳式程序综合 (inductive program synthesis)：归纳式程序综合基于归纳式规约，例如输入输出对、示例 (demonstration) 等。

线性时序逻辑（Linear Temporal Logic，LTL）

- LTL 的语法定义如下：
  - ☞ 给定原子命题集合 AP

$$\varphi ::= p \in AP \mid \neg\varphi \mid \varphi_1 \vee \varphi_2 \mid \bigcirc \varphi \mid \varphi_1 \mathcal{U} \varphi_2$$

  - ☞ 派生算子

    常量：

$$True \equiv p \vee \neg p \quad False \equiv \neg True$$

    命题逻辑算子：

$$\varphi_1 \wedge \varphi_2 \equiv \neg(\neg\varphi_1 \vee \neg\varphi_2) \quad \varphi_1 \rightarrow \varphi_2 \equiv \neg\varphi_1 \vee \varphi_2$$

$$\varphi_1 \leftrightarrow \varphi_2 \equiv (\varphi_1 \rightarrow \varphi_2) \wedge (\varphi_2 \rightarrow \varphi_1)$$

    时序算子：

$$\Diamond\varphi \equiv True \mathcal{U} \varphi \quad \Box\varphi \equiv \neg\Diamond\neg\varphi$$

线性时序逻辑（Linear Temporal Logic，LTL）

- 给定由 AP 的子集构成的无穷序列 π 和一个位置 $i \in \mathbb{N}$，用 $\pi(i)$ 表示 π 上的第 i 个元素（$\pi(i) \in 2^{AP}$），π、i 和 LTL 公式之间的满足关系定义如下：

  - ☞ $\pi, i \models p$ iff $p \in \pi(i)$
  - ☞ $\pi, i \models \neg\varphi$ iff $\pi, i \not\models \varphi$
  - ☞ $\pi, i \models \varphi_1 \vee \varphi_2$ iff $\pi, i \models \varphi_1$ 或 $\pi, i \models \varphi_2$
  - ☞ $\pi, i \models \bigcirc\varphi$ iff $\pi, i+1 \models \varphi$
  - ☞ $\pi, i \models \varphi_1 \mathcal{U} \varphi_2$ iff $\exists k \geqslant i$ 使得 $\pi, k \models \varphi_2$ 且 $\forall i \leqslant j < k$ 都有 $\pi, j \models \varphi_1$

  - ☞ $\pi, i \models \Diamond\varphi$ iff $\exists k \geqslant i$ 使得 $\pi, k \models \varphi$
  - ☞ $\pi, i \models \Box\varphi$ iff $\forall k \geqslant i$ 都有 $\pi, k \models \varphi$
  - ☞ $\pi, i \models \Box\Diamond\varphi$ iff $\forall k \geqslant i$ 都有 $\exists j \geqslant k$ 使得 $\pi, j \models \varphi$

公平离散系统（Fair Discrete System，FDS）

- 一个 FDS $D =< \mathcal{V}, \theta, \rho, \mathcal{J}, \mathcal{C} >$ 由以下元素组成：
  - ☞ $\mathcal{V} = \{v_1, \ldots, v_n\}$：布尔变量的有限集合，一个状态 s 是对 $\mathcal{V}$ 的一个指派，即 $s \in \Sigma_{\mathcal{V}} = 2^{\mathcal{V}}$；
  - ☞ θ：初始条件，定义在 $\mathcal{V}$ 上的断言，满足 θ 的状态被称为初始状态；
  - ☞ ρ：迁移关系，定义在 $\mathcal{V} \cup \mathcal{V}'$ 上的断言；
  - ☞ $\mathcal{J} = \{J_1, \ldots, J_m\}$：弱公平性条件集合，$J \in \mathcal{J}$ 是定义在 $\mathcal{V}$ 上的断言；
  - ☞ $\mathcal{C} = \{(P_1, Q_1), \ldots, (P_n, Q_N)\}$：强公平性条件集合，$(P, Q) \in \mathcal{C}$ 是一对定义在 $\mathcal{V}$ 上的断言。

- $\mathcal{J}$ 和 $\mathcal{C}$ 均为 $\emptyset$ 的 FDS 被称为无公平性约束（fairness-free）的 FDS。

公平离散系统（Fair Discrete System，FDS）

- FDS 的一个运行（run）是满足下面两个条件的极大（状态）序列 $\sigma = s_0, s_1, \ldots$：
  - ☞ 初始性，即 $s_0 \models \theta$; ☞ 连续性，即对所有的 $j \geqslant 0$，都有 $(s_j, s_{j+1}) \models \rho$。
- 极大序列指：
  - ☞ 序列长度是无限的，
  - ☞ 或者序列长度是有限的，且最后一个状态不存在符合 ρ 的后继。
- $\mathcal{D}$ 是一个无公平性约束的 FDS，如果 $\mathcal{D}$ 的所有运行都是无限运行，且满足规约 φ，则称 $\mathcal{D}$ 满足（implements）φ，记作 $\mathcal{D} \models \varphi$。
- 满足以下两个条件，称 $\mathcal{D}$ 对 $\mathcal{X} \subseteq \mathcal{V}$ 是完全的：
  - ☞ 对于 $\mathcal{X}$ 的所有指派 $s_{\mathcal{X}} \in \Sigma_{\mathcal{X}}$，都存在状态 $s \in \Sigma_{\mathcal{V}}$ 使得 $s|_{\mathcal{X}} = s_{\mathcal{X}}$ 且 $s \models \theta$;
  - ☞ 对于 $\mathcal{X}$ 的所有指派 $s'_{\mathcal{X}} \in \Sigma_{\mathcal{X}}$ 以及所有状态 $s \in \Sigma_{\mathcal{V}}$，都存在状态 $s' \in \Sigma_{\mathcal{V}}$，使得 $s'|_{\mathcal{X}} = s'_{\mathcal{X}}$ 且 $(s, s') \models \rho$。

可实现性与综合（Realizability and Synthesis）

给定一个无公平性约束的 FDS $\mathcal{D} =< \mathcal{V}, \theta, \rho >$，一个定义在输入变量 $\mathcal{X}$ 和输出变量 $\mathcal{Y}$ 上的 LTL 公式 φ，

- 符合以下条件，则称 $\mathcal{D}$ 实现（realize）了 φ，并称 $\mathcal{D}$ 是 φ 的控制器：
  - ☞ $\mathcal{V}$ 包含 $\mathcal{X}$ 和 $\mathcal{Y}$；
  - ☞ $\mathcal{D}$ 对 $\mathcal{X}$ 是完全的；
  - ☞ $\mathcal{D} \models \varphi$。
- 对于规约 φ，如果存在一个 $\mathcal{D}$ 可以实现 φ，则称 φ 是可实现的（realizable），否则是不可实现的（unrealizable）。为 φ 构建控制器 $\mathcal{D}$ 的问题称为综合问题（synthesis problem）。

博弈结构（Game Structure）

系统和环境之间的双人博弈，系统目标是无论环境如何，都要使规约被满足。

- 一个博弈结构 $G =< \mathcal{V}, \mathcal{X}, \mathcal{Y}, \theta_e, \theta_s, \rho_e, \rho_s, \varphi >$ 由以下元素组成：
  - ☞ $\mathcal{V} = \{v_1, \ldots, v_n\}$：布尔变量的有限集合，一个状态 s 是对 $\mathcal{V}$ 的一个指派，即 $s \in \Sigma_{\mathcal{V}} = 2^{\mathcal{V}}$；
  - ☞ $\mathcal{X} \subseteq \mathcal{V}$ 是输入变量集合，即环境控制的变量；
  - ☞ $\mathcal{Y} = \mathcal{V} \setminus \mathcal{X}$ 是输出变量集合，即系统控制的变量；
  - ☞ θ_e：定义在 $\mathcal{X}$ 上的断言，对初始状态的环境部分进行限制；
  - ☞ θ_s：定义在 $\mathcal{V}$ 上的断言，对初始状态的系统部分进行限制；
  - ☞ $\rho_e(\mathcal{V}, \mathcal{X}')$：环境迁移关系，对一个状态和它下一步可能的输入之间的关系进行了限制；
  - ☞ $\rho_s(\mathcal{V}, \mathcal{X}', \mathcal{Y}')$：系统迁移关系，对一个状态和下一步输入及下一步输出之间的关系进行了限制；
  - ☞ φ：获胜条件，形式为一个 LTL 公式。

博弈结构（Game Structure）

- 博弈结构 $G =< \mathcal{V}, \mathcal{X}, \mathcal{Y}, \theta_e, \theta_s, \rho_e, \rho_s, \varphi >$ 的一次博弈（play）是满足下面两个条件的极大（状态）序列 $\sigma = s_0, s_1, \ldots$：
  - ☞ 初始性，即 $s_0 \models \theta_e \wedge \theta_s$;
  - ☞ 连续性，即对所有的 $j \geqslant 0$，都有 $(s_j, s_{j+1}) \models \rho_e \wedge \rho_s$。
- 一次博弈 $\sigma = s_0, s_1, \ldots$ 是系统获胜的（winning for system），如果：
  - ☞ σ 长度是有限的，且对于最后一个状态 s_n，不存在赋值 $s_{\mathcal{X}} \in \Sigma_{\mathcal{X}}$ 使得 $(s_n, s_{\mathcal{X}}) \models \rho_e$，
  - ☞ 或者 σ 长度是无限的，且 $\sigma \models \varphi$。
- 系统的（无记忆）策略（(memoryless) strategy）是一个函数 $f : \Sigma_{\mathcal{V}} \times \Sigma_{\mathcal{X}} \mapsto \Sigma_{\mathcal{Y}}$，使得对于任意 $s \in \Sigma_{\mathcal{V}}$ 和 $s_{\mathcal{X}} \in \Sigma_{\mathcal{X}}$，如果 $(s, s_{\mathcal{X}}) \models \rho_e$，则 $(s, s_{\mathcal{X}}, f(s, s_{\mathcal{X}})) \models \rho_s$。

博弈结构（Game Structure）

- 给定博弈 $\sigma = s_0, s_1, \ldots$ 和系统策略 f，如果对于所有 $i \geqslant 0$ 都有 $f(s_i, s_{i+1}|_{\mathcal{X}}) = (s_{i+1}|_{\mathcal{Y}})$，则称 σ 是符合（compliant with）f 的。
- 如果所有从状态 $s \in \Sigma_{\mathcal{V}}$ 出发，且符合策略 f 的博弈，均是系统获胜的，则称 f 从 s 出发是系统获胜的。
- 将存在系统获胜策略的状态称为系统获胜状态，记作 W_s。
- 环境策略、环境获胜策略、环境获胜状态 W_e 均可对偶定义。
- 如果对于所有 $s_x \in \Sigma_{\mathcal{X}}$ 且 $s_x \models \theta_e$，都存在 $s_{\mathcal{Y}} \in \Sigma_{\mathcal{Y}}$ 使得 $(s_{\mathcal{X}}, s_{\mathcal{Y}}) \models \theta_s$ 且 $(s_{\mathcal{X}}, s_{\mathcal{Y}}) \models W_s$，则称这个博弈结构是系统获胜的。

定理：给定博弈结构 $G =< \mathcal{V}, \mathcal{X}, \mathcal{Y}, \theta_e, \theta_s, \rho_e, \rho_s, \varphi >$，和公式

$$\varphi_G = (\theta_e \to \theta_s) \wedge (\theta_e \to \square((\boxminus \rho_e) \to \rho_s)) \wedge (\theta_e \wedge \square\rho_e \to \varphi),$$

φ_G 是可实现的当且仅当 G 是系统获胜的。

LTL 的综合问题

- 对 LTL 的综合问题及反应式综合问题的研究开始于上世纪八十年代，LTL 的反应式系统综合需要为 LTL 规约依次构建Büchi 自动机和Rabin 自动机，时间复杂度达到双指数。
- 为降低复杂度，构建获胜条件为$\Box\varphi$、$\Diamond\varphi$、$\Box\Diamond\varphi$、$\Diamond\Box\varphi$的博弈，存在高效的求解算法。
- 构建获胜条件为Generalized Reactivity(1) 公式的博弈，使得规约同时兼具表达能力强和求解算法高效的优点。

$$\varphi_{gr} = (\Box\Diamond p_1 \wedge \cdots \wedge \Box\Diamond p_m) \rightarrow (\Box\Diamond q_1 \wedge \cdots \wedge \Box\Diamond q_n)$$

GR(1) 规约

- 给定环境命题集合 $\mathcal{X}$ 和系统命题集合 $\mathcal{Y}$，GR(1) 规约由六个部分组成：
  环境约束：
  - ☞ φ_i^e 约束环境初始值，是由 $\mathcal{X}$ 中的命题构成的布尔公式，不含时序算子；
  - ☞ φ_t^e 约束环境状态迁移，形如 $\bigwedge \Box B_i$，B_i 是定义在 $\mathcal{X} \cup \mathcal{Y} \cup \bigcirc \mathcal{X}$ 上的布尔公式；
  - ☞ φ_g^e 约束环境目标，形如 $\bigwedge \Box \Diamond B_i$，$B_i$ 是定义在 $\mathcal{X} \cup \mathcal{Y}$ 上的布尔公式；

  系统约束：
  - ☞ φ_i^s 约束系统初始值，是由 $\mathcal{Y}$ 中的命题构成的布尔公式，不含时序算子；
  - ☞ φ_t^s 约束系统状态迁移，形如 $\bigwedge \Box B_i$，B_i 是定义在 $\mathcal{X} \cup \mathcal{Y} \cup \bigcirc \mathcal{X} \cup \bigcirc \mathcal{Y}$ 上的布尔公式；
  - ☞ φ_g^s 约束系统目标，形如 $\bigwedge \Box \Diamond B_i$，$B_i$ 是定义在 $\mathcal{X} \cup \mathcal{Y}$ 上的布尔公式；
- $$\varphi_{e,s}^{\rightarrow} = (\varphi_i^e \wedge \varphi_t^e \wedge \varphi_g^e) \rightarrow (\varphi_i^s \wedge \varphi_t^s \wedge \varphi_g^s)$$

GR(1) 规约

- 令 $\theta_e = \varphi_i^e$，$\theta_s = \varphi_i^s$，ρ_e 和 ρ_s 分别为 φ_t^e 和 φ_t^s 去除 $\Box$ 后的变化形式，$\varphi = \varphi_g^e \to \varphi_g^s$，则可以构建博弈结构 $G =< \mathcal{V}, \mathcal{X}, \mathcal{Y}, \theta_e, \theta_s, \rho_e, \rho_s, \varphi >$，有以下两个公式：
  - ☞ $\varphi_G = (\theta_e \to \theta_s) \wedge (\theta_e \to \Box((\boxminus \rho_e) \to \rho_s)) \wedge (\theta_e \wedge \Box\rho_e \to \varphi)$
  - ☞ $\varphi_{e,s}^{\to} = (\varphi_i^e \wedge \varphi_t^e \wedge \varphi_g^e) \to (\varphi_i^s \wedge \varphi_t^s \wedge \varphi_g^s)$

定理：$\varphi_G \to \varphi_{e,s}^{\to}$，实现 φ_G 的控制器也可以实现 $\varphi_{e,s}^{\to}$，反之不成立。

GR(1) 规约

- 命题
  - ☞ $\mathcal{X} = \{s^{Nemo}\}$ 发现 Nemo
  - ☞ $\mathcal{Y} = \{r_1, \ldots r_{12}, a^{CameraOn}\}$ 12 个区域及“打开照相机”
- 规约
  - ☞ 只能在区域 1、3、5、8 找到 Nemo;
  - ☞ 在区域 1、3、5、8 寻找 Nemo;
  - ☞ 如果找到，打开照相机并停在原地;
  - ☞ 如果 Nemo 消失，继续寻找;
  - ☞ 出发区域为 1、2、3，初始状态下 Nemo 不出现。

GR(1) 规约

- 规约
  - ☞ $\varphi_i^e : \neg s^{Nemo}$
  - ☞ $\varphi_t^e : \Box((\neg r_1 \wedge \neg r_3 \wedge \neg r_5 \wedge \neg r_8) \rightarrow (\neg \bigcirc s^{Nemo}))$
  - ☞ $\varphi_g^e : \Box\Diamond True$
  - ☞ $\varphi_i^s : (r_1 \vee r_2 \vee r_3) \wedge (\neg a^{CameraOn})$
  - ☞ $\varphi_t^s : \Box(\bigcirc s^{Nemo} \rightarrow (\bigwedge_{i\in\{1,\ldots,12\}} \bigcirc r_i \leftrightarrow r_i) \wedge \bigcirc a^{CameraOn})$
    $\wedge\Box(\neg \bigcirc s^{Nemo} \rightarrow \neg \bigcirc a^{CameraOn})$
  - ☞ $\varphi_g^s : \Box\Diamond(r_1 \vee s^{Nemo}) \wedge \Box\Diamond(r_3 \vee s^{Nemo}) \wedge \Box\Diamond(r_5 \vee s^{Nemo}) \wedge \Box\Diamond(r_8 \vee s^{Nemo})$

GR(1) 规约

- 规约

ϕ_i^e: $(\neg s^{\text{Nemo}})$ ϕ_t^e:$\square((\neg r_1 \wedge \neg r_3 \wedge \neg r_5 \wedge \neg r_8) \rightarrow (\neg \bigcirc s^{\text{Nemo}}))$

ϕ_g^e:$\square \diamond (True)$

ϕ_i^s: $(r_1 \vee r_2 \vee r_3) \wedge (\neg a^{\text{CameraOn}})$

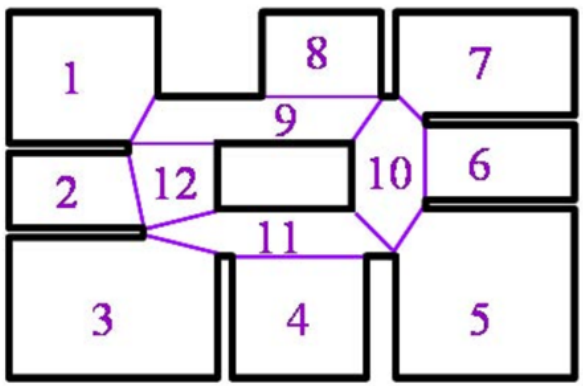
ϕ_t^s:$\square\big(\bigcirc s^{\text{Nemo}} \rightarrow (\wedge_{i\in\{1,\dots,12\}} \bigcirc r_i \leftrightarrow r_i) \wedge \bigcirc a^{\text{CameraOn}}\big)$

$\wedge \square(\neg \bigcirc s^{\text{Nemo}} \rightarrow \neg \bigcirc a^{\text{CameraOn}})$

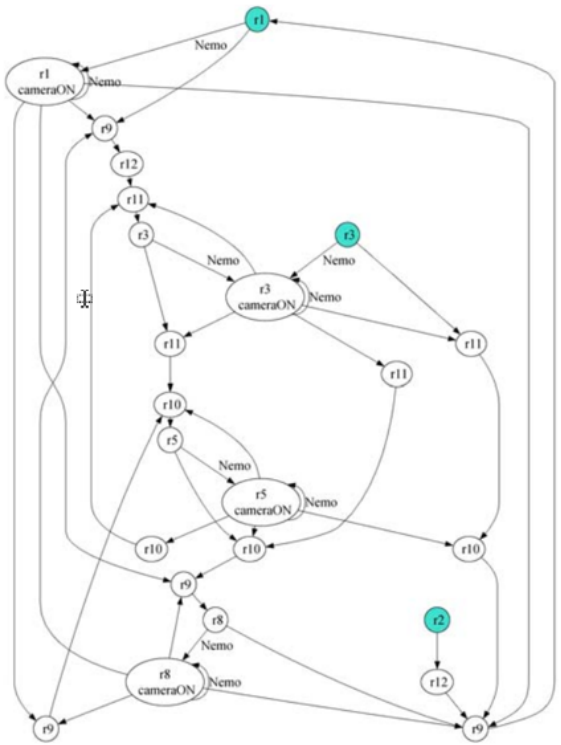
ϕ_g^s:$\square \diamond (r_1 \vee s^{\text{Nemo}}) \wedge \square \diamond (r_3 \vee s^{\text{Nemo}}) \wedge \square \diamond (r_5 \vee s^{\text{Nemo}})$

$\wedge \square \diamond (r_8 \vee s^{\text{Nemo}})$

- 地图



- 控制器



博弈结构上的 μ 演算

- 给定博弈结构 $G =< \mathcal{V}, \mathcal{X}, \mathcal{Y}, \theta_e, \theta_s, \rho_e, \rho_s, \varphi >$，关系变量集合 $Var = \{X, Y, \dots\}$，μ 演算公式语法定义如下：

$$\varphi \ ::= \ v \in \mathcal{V} \mid \neg v \mid X \mid \varphi_1 \vee \varphi_2 \mid \varphi_1 \wedge \varphi_2 \mid \langle\!\langle \rangle\!\rangle \varphi \mid [\![ ]\!] \varphi \mid \mu X \varphi \mid \nu X \varphi$$

博弈结构上的 μ 演算

- μ 演算公式 ψ 的语义解释为 G 上的状态集合，即 ψ 为真的那些状态，写作 $[[\psi]]^{\mathcal{E}}_G$，其中 $\mathcal{E} : Var \to 2^{\Sigma}$ 是环境，将状态集合赋值给关系变量。
- μ 演算公式语义定义如下：
  - ☞ $[[v]]^{\mathcal{E}}_G = \{s \in \Sigma | s[v] = 1\}$；
  - ☞ $[[\neg v]]^{\mathcal{E}}_G = \{s \in \Sigma | s[v] = 0\}$；
  - ☞ $[[X]]^{\mathcal{E}}_G = \mathcal{E}(X)$；
  - ☞ $[[\varphi \vee \psi]]^{\mathcal{E}}_G = [[\varphi]]^{\mathcal{E}}_G \cup [[\psi]]^{\mathcal{E}}_G$；
  - ☞ $[[\varphi \wedge \psi]]^{\mathcal{E}}_G = [[\varphi]]^{\mathcal{E}}_G \cap [[\psi]]^{\mathcal{E}}_G$；
  - ☞ $[[\Diamond \varphi]]^{\mathcal{E}}_G = \{s \in \Sigma | \forall s_{\mathcal{X}} \in \Sigma_{\mathcal{X}}, (s, s_{\mathcal{X}}) \models \rho_e \to \exists s_{\mathcal{Y}} \in \Sigma_{\mathcal{Y}}$ 使得 $(s, s_{\mathcal{X}}, s_{\mathcal{Y}}) \models \rho_s$ 且 $(s_{\mathcal{X}}, s_{\mathcal{Y}}) \in [[\varphi]]^{\mathcal{E}}_G\}$
  - ☞ $[[\Box \varphi]]^{\mathcal{E}}_G = \{s \in \Sigma | \exists s_{\mathcal{X}} \in \Sigma_{\mathcal{X}}$ 使得 $(s, s_{\mathcal{X}}) \models \rho_e$ 且 $\forall s_{\mathcal{Y}} \in \Sigma_{\mathcal{Y}}, (s, s_{\mathcal{X}}, s_{\mathcal{Y}}) \models \rho_s \to (s_{\mathcal{X}}, s_{\mathcal{Y}}) \in [[\varphi]]^{\mathcal{E}}_G\}$
  - ☞ $[[\mu X \varphi]]^{\mathcal{E}}_G = \bigcup_i S_i$，其中 $S_0 = \emptyset$ 且 $S_{i+1} = [[\varphi]]^{\mathcal{E}[X \leftarrow S_i]}_G$；
  - ☞ $[[\nu X \varphi]]^{\mathcal{E}}_G = \bigcap_i S_i$，其中 $S_0 = \Sigma$ 且 $S_{i+1} = [[\varphi]]^{\mathcal{E}[X \leftarrow S_i]}_G$；

博弈结构上的 μ 演算

- 如何计算 $[[\mu X\varphi]]^{\mathcal{E}}_G$（或 $[[\nu X\varphi]]^{\mathcal{E}}_G$）?
  - ☞ 1. 令 $S_0 = \emptyset$（或 $S_0 = \Sigma$），
  - ☞ 2. 令 $S_{i+1} = [[\varphi]]^{\mathcal{E}[X \leftarrow S_i]}_G$，
  - ☞ 3. 一旦 $S_{l+1} = S_l$，即停止计算，令 $[[\mu X\varphi]]^{\mathcal{E}}_G = S_l$（或 $[[[\nu X\varphi]]^{\mathcal{E}}_G = S_l$）。

GR(1) 公式

- LTL 形式

$$\begin{aligned}\varphi &= \varphi_g^e \rightarrow \varphi_g^s \\ &= \bigwedge_{i=1}^{m} \square\diamond J_i^e \rightarrow \bigwedge_{j=1}^{n} \square\diamond J_j^s\end{aligned} \tag{1}$$

- 博弈结构上的 μ 演算形式

$$\varphi_{gr} = \nu Z(\bigwedge_{j=1}^{n} \mu Y(\bigvee_{i=1}^{m} \nu X(J_j^s \wedge \bigcirc Z \vee \bigcirc Y \vee \neg J_i^e \wedge \bigcirc X)))$$

$$= \nu \begin{bmatrix} Z_1 \\ Z_2 \\ \vdots \\ \vdots \\ Z_n \end{bmatrix} \begin{bmatrix} \mu Y(\bigvee_{i=1}^{m} \nu X(J_1^s \wedge \bigcirc Z_2 \vee \bigcirc Y \vee \neg J_i^e \wedge \bigcirc X)) \\ \mu Y(\bigvee_{i=1}^{m} \nu X(J_2^s \wedge \bigcirc Z_3 \vee \bigcirc Y \vee \neg J_i^e \wedge \bigcirc X)) \\ \vdots \\ \vdots \\ \mu Y(\bigvee_{i=1}^{m} \nu X(J_n^s \wedge \bigcirc Z_1 \vee \bigcirc Y \vee \neg J_i^e \wedge \bigcirc X)) \end{bmatrix} \tag{2}$$

GR(1) 公式

- 博弈结构上的 μ 演算形式

$$\nu Z(\bigwedge_{j=1}^{n} \mu Y(\bigvee_{i=1}^{m} \nu X(J_j^s \wedge \bigcirc Z \vee \bigcirc Y \vee \neg J_i^e \wedge \bigcirc X)))$$

- 核心

$$J_j^s \wedge \bigcirc Z \vee \bigcirc Y \vee \neg J_i^e \wedge \bigcirc X$$

- 初始状态 $Z = T\ Y = F\ X = T$

$$J_j^s \vee \neg J_i^e$$

- 对 $\nu X(J_j^s \vee \neg J_i^e \wedge \bigcirc X)$ 迭代

$$J_j^s \vee \neg J_i^e$$

$$J_j^s \vee \neg J_i^e \wedge \bigcirc (J_j^s \vee \neg J_i^e)$$

$$J_j^s \vee \neg J_i^e \wedge \bigcirc (J_j^s \vee \neg J_i^e \wedge \bigcirc (J_j^s \vee \neg J_i^e))$$

$$J_j^s \vee \neg J_i^e \wedge \cdots \bigcirc (J_j^s \vee \neg J_i^e \wedge \bigcirc (J_j^s \vee \neg J_i^e)) \ -- fix\ point$$

GR(1) 公式

- 可得 $\nu X(J_j^s \vee \neg J_i^e \wedge \bigcirc X)$ 的语义

$$J_j^s \vee \neg J_i^e \wedge \cdots \bigcirc (J_j^s \vee \neg J_i^e \wedge \bigcirc (J_j^s \vee \neg J_i^e))$$

系统存在一个策略，迫使博弈

$$\neg J_i^e \, W J_j^s = \begin{cases} \text{i) 在有限步数内抵达} J_j^s, \text{且在此之前保持} \neg J_i^e; \\ \text{ii) 一直保持} \neg J_i^e. \end{cases}$$

- $\bigvee_{i=1}^m \nu X(J_j^s \vee \neg J_i^e \wedge \bigcirc X)$ 的语义

系统存在一个策略，迫使博弈

$$\bigvee_{i=1}^m \neg J_i^e \, W J_j^s = \begin{cases} \text{i) 在有限步数内抵达} J_j^s, \text{且在此之前保持} \neg J_i^e, i \in [1, m]; \\ \text{ii) 一直保持} \neg J_i^e, i \in [1, m]. \end{cases}$$

GR(1) 公式

- 对 $\mu Y(\bigvee_{i=1}^{m} \nu X(J_j^s \vee \bigcirc Y \vee \neg J_i^e \wedge \bigcirc X))$ 迭代

$$Y = F, Q = \begin{cases} \text{i) 有限步数内抵达} J_j^s, \text{且在此之前保持} \neg J_i^e, i \in [1, m]; \\ \text{ii) 一直保持} \neg J_i^e, i \in [1, m]. \end{cases}$$

$$Y = Q, Q' = \begin{cases} \text{i) 有限步数内抵达} J_j^s \cup \bigcirc Q, \text{且在此之前保持} \neg J_i^e, i \in [1, m]; \\ \text{ii) 一直保持} \neg J_i^e, i \in [1, m]. \end{cases}$$

$$Y = Q', Q'' = \begin{cases} \text{i) 有限步数内抵达} J_j^s \cup \bigcirc Q', \text{且在此之前保持} \neg J_i^e, i \in [1, m]; \\ \text{ii) 一直保持} \neg J_i^e, i \in [1, m]. \end{cases}$$

- $\mu Y(\bigvee_{i=1}^{m} \nu X(J_j^s \vee \bigcirc Y \vee \neg J_i^e \wedge \bigcirc X))$ 语义

$$\text{系统存在一个策略，迫使博弈} \begin{cases} \text{i) 在有限步数内抵达} J_j^s; \\ \text{ii) 在有限步数内开始一直保持} \neg J_i^e, i \in [1, m]. \end{cases}$$

$$Q_Y = (\bigvee_{i=1}^{m} \Diamond\Box \neg J_i^e) \vee \Diamond J_j^s = \bigwedge_{i=1}^{m} \Box\Diamond J_i^e \rightarrow \Diamond J_j^s$$

GR(1) 公式

- $\bigwedge_{j=1}^{n} \mu Y(\bigvee_{i=1}^{m} \nu X(J_j^s \vee \bigcirc\, Y \vee \neg J_i^e \wedge \bigcirc\, X))$ 语义

  系统存在一个策略，迫使博弈 $\begin{cases} \text{i) 在有限步数内抵达的状态覆盖} J_1^s \dots J_n^s; \\ \text{ii) 在有限步数内开始一直保持} \neg J_i^e, i \in [1, m]. \end{cases}$

$$\bigwedge_{i=1}^{m} \Box\Diamond J_i^e \to \bigwedge_{j=1}^{n} \Diamond J_j^s$$

- 对 $\nu Z(\bigwedge_{j=1}^{n} \mu Y(\bigvee_{i=1}^{m} \nu X(J_j^s \wedge \bigcirc\, Z \vee \bigcirc\, Y \vee \neg J_i^e \wedge \bigcirc\, X)))$ 迭代

$$Z = T, Q = \begin{cases} \text{i) 在有限步数内抵达的状态覆盖} J_1^s \dots J_n^s; \\ \text{ii) 在有限步数内开始一直保持} \neg J_i^e, i \in [1, m]. \end{cases}$$

$$Z = Q, Q' = \begin{cases} \text{i) 在有限步数内抵达的状态覆盖} J_1^s \wedge \bigcirc\, Q \dots J_n^s \wedge \bigcirc\, Q; \\ \text{ii) 在有限步数内开始一直保持} \neg J_i^e, i \in [1, m]. \end{cases}$$

$$Z = Q', Q'' = \begin{cases} \text{i) 在有限步数内抵达的状态覆盖} J_1^s \wedge \bigcirc\, Q' \dots J_n^s \wedge \bigcirc\, Q'; \\ \text{ii) 在有限步数内开始一直保持} \neg J_i^e, i \in [1, m]. \end{cases}$$

GR(1) 公式

- $\nu Z(\bigwedge_{j=1}^{n} \mu Y(\bigvee_{i=1}^{m} \nu X(J_j^s \wedge \bigcirc Z \vee \bigcirc Y \vee \neg J_i^e \wedge \bigcirc X)))$ 语义

  系统存在一个策略，迫使博弈

$$\begin{cases} \text{i) 无限次覆盖} J_1^s \dots J_n^s; \\ \text{ii) 在有限步数内开始一直保持} \neg J_i^e, i \in [1, m]. \end{cases}$$

$$\bigwedge_{i=1}^{m} \Box\Diamond J_i^e \rightarrow \bigwedge_{j=1}^{n} \Box\Diamond J_j^s$$

GR(1) 公式——Jtlv 实现

```
public BDD calculate_win() {
   BDD Z = TRUE;
   for (Fix fZ = new Fix(); fZ.advance(Z);) {
      mem.clear();
      for (int j = 1; j <= sys.numJ(); j++) {
         BDD Y = FALSE;
         for (Fix fY = new Fix(); fY.advance(Y);) {
            BDD start = sys.Ji(j).and(cox(Z)).or(cox(Y));
            Y = FALSE;
            for (int i = 1; i <= env.numJ(); i++) {
               BDD X = Z;
               for (Fix fX = new Fix(); fX.advance(X);)
                  X = start.or(env.Ji(i).not().and(cox(X)));
               mem.addX(j, i, X); // store values of X
               Y = Y.or(X);
            }
            mem.addY(j, Y); // store values of Y
         }
         Z = Y;
      }
   }
   return Z;
}
```

构造策略

- $mY[j][r]$ 集合的语义：

$$\text{系统可以迫使博弈}\begin{cases}\text{i) 在 } r \text{ 步数内抵达} J_j^s \wedge \bigcirc Z;\\ \text{ii) 一直保持} \neg J_i^e, i \in [1, m].\end{cases}$$

- $mX[j][r][i]$ 集合的语义：

$$\text{系统可以迫使博弈}\begin{cases}\text{i) 在 } r \text{ 步数内抵达} J_j^s \wedge \bigcirc Z;\\ \text{ii) 一直保持} \neg J_i^e.\end{cases}$$

- 构造策略的思路：构建子策略 f_j，迫使博弈在有限步内抵达 J_j^s，或持续违反某个 J_i^e。
  - ☞ $mY[j][r]$ 向 r 变小的方向取值，可以最终抵达 J_j^s，然后可以切换策略到 $f_{j\oplus 1}$；
  - ☞ $mX[j][r][i]$ 保持 r 不变小，可以一直违反 J_i^e；
  - ☞ 将所有子策略结合起来，可以无限多次满足系统目标，或持续违反某个环境目标。

构造策略

- 由博弈结构 $G =< \mathcal{V}, \mathcal{X}, \mathcal{Y}, \theta_e, \theta_s, \rho_e, \rho_s, \varphi >$，构造 FDS $D =< \mathcal{V}_{\mathcal{D}}, \theta_{\mathcal{D}}, \rho >$
  - ☞ $\mathcal{V}_{\mathcal{D}} = \mathcal{V} \cup \mathcal{Z}_n$，
  - ☞ $\theta_{\mathcal{D}} = \theta_e \rightarrow (\theta_s \wedge \mathcal{Z}_n = 1 \wedge z)$，
  - ☞ $\rho = (\rho_e \wedge z) \rightarrow (\rho_1 \wedge \rho_2 \wedge \rho_3)$。
- 迁移定义

$$\rho_1 = \bigvee_{j \in [1..n]} (\mathcal{Z}_n = j) \wedge J_j^s \wedge \rho_s \wedge z' \wedge (\mathcal{Z}'_n = j \oplus 1)$$

$$\rho_2(j) = \bigvee_{r>1} mY[j][r] \wedge \neg mY[j][< r] \wedge \rho_s \wedge mY'[j][< r]$$

$$\rho_2 = \bigvee_{j \in [1..n]} (\mathcal{Z}_n = \mathcal{Z}'_n = j) \wedge \rho_2(j)$$

$$\rho_3(j) = \bigvee_{r} \bigvee_{i \in [1..m]} mX[j][r][i] \wedge \neg mX[j][\prec (r, i)] \wedge \neg J_i^e \wedge \rho_s \wedge mX'[j][r][i]$$

$$\rho_3 = \bigvee (\mathcal{Z}_n = \mathcal{Z}'_n = j) \wedge \rho_3(j)$$

谢　谢!

Thank you!