

Logique Temporelle

Cours de master M1 IA

L. Kahloul

Plan

- Logique: C'est quoi, éléments, syntaxe, sémantique;
- Types: Propositionnelles, des prédicats (1^{er} ordre), limites des ces logique;
- Logique temporelle (LT): quoi de neuf?, syntaxe sémantique;
- Classes de LT: LTL, CTL*, CTL

La logique (informellement)

- Un langage pour la spécification et la représentation des connaissances
- Un langage= alphabet, mots, règles pour former des phrases (formules bien formées: **syntaxe**)
- Un langage formel: **Syntaxe+Sémantique** (les deux formellement définies)
- Classes : logique **propositionnelle**, logique de **prédicats**, logiques **modales**,...

Logique des Propositions

- Langage :
Mots de base = variables propositionnelle $V=\{P, Q, R, \dots\}$, connecteurs $\{\neg, \wedge, \vee\}$, parenthèse $\{(,)\}$
- Syntaxe: **règles pour former** les **formules** appartenant à LP, exemple :
 $(P \wedge Q)$ est une formule dans LP
$$\varphi ::= \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \neg \varphi \mid p$$
- Sémantique : définition **d'un modèle** M. Un modèle est un couple (D, π) , où D est **un domaine**, et π est **une application (interprétation)** qui associe à chaque variable propositionnelle une valeur dans D.

Pour LP:

$$D=\{0,1\}$$

$$\pi:V \rightarrow D$$

Logique des Propositions

- Une formule φ est dite **satisfaite** dans un modèle M , et on note ça par $M \models \varphi$ ssi φ est **évaluée à 1 dans M**
- φ est valide (**tautologie**) ssi elle est satisfaite dans **tous les modèles**
- φ est une **antilogie (contradiction)** ssi **aucun modèle M** ne satisfait φ

Une **théorie** est un ensemble de formules. On reprend les mêmes remarques pour une théorie

Logique des Prédicats

- Une **extension** de la logique des propositionnelle:
 - 1) les propositions ne sont plus des variables mais **des fonctions avec arguments**, donc des **Prédicats**
 - 2) introduire des **constantes**, des **variables** et des **fonctions** non propositionnelles
 - 3) introduire des **quantificateurs** sur les variables non propositionnelles
- Appelé aussi Logique d'ordre 1: les prédicats peuvent porter seulement sur des « **variables** ou **des fonctions** » **non propositionnelles**.

Logique des Prédicats

- Langage :

Partie fixe:

connecteurs $\{\neg, \wedge, \vee, \Leftrightarrow, \Rightarrow\}$,

parenthèse $\{(,)\}$

Quantificateurs $=\{\forall, \exists\}$

Partie variable:

Un ensemble de constantes $C=\{a, b, c, \dots\}$

Un ensemble de variables $V=\{x, y, \dots\}$

Un ensemble de fonctions $F=\{f_1, f_2, \dots\}$

Un ensemble de Prédicats $P=\{P_1, P_2, \dots\}$

Ces deux logiques sont elles
suffisante?

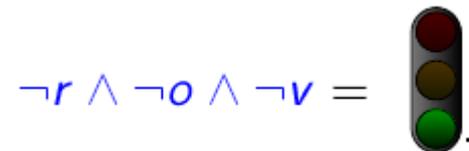
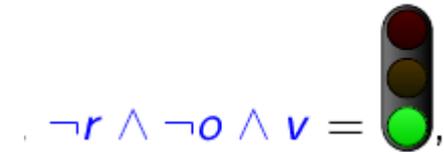
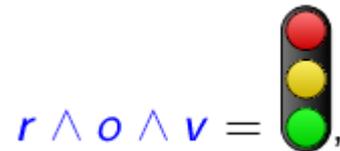
On peut faire avec :

- Exprime des propriétés sur l'instant, exemple :

r : feu rouge allumé

o : feu orange allumé

v : feu vert allumé



On ne pas peut faire avec !!!!!

Comment dire que  précède  ?

Comment dire que le système ne reste pas toujours sur  ?

⇒ besoin de faire apparaître le temps

Introduction : besoin d'une LT

- 1) Logique classique + **Opérateur dédiés au temps (Opérateurs temporels)**
- 2) **Opérateurs temporels**: permettant d'exprimer l'évolution d'un système au cours du temps, tels que : ***toujours, jusqu'à, fatalement*** Associent **une valeur de vérité non pas à un état**, mais à **une séquence d'états**, représentant l'évolution d'un système au cours du temps
- 3) Deux sortes d'opérateurs : **Connecteur temporels** + **quantificateurs de chemins**
- 4) Classes de logiques temporelles:
 - Les logiques arborescentes (CTL*,CTL).
 - Les logiques linéaires (LTL).

Logique Temporelle

connecteurs et quantificateurs

Les connecteurs :

- Expriment une propriété sur un seul chemin: suite **d'événements attendus** sur un seul chemin
- 4 connecteurs sont définis : X, F, G, U

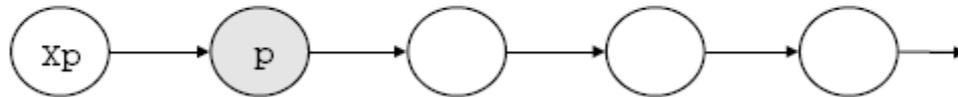
Les quantificateurs:

- Exprime une propriété sur un **dépliage** (probablement plusieurs chemins à partir d'un certain état)
- 2 quantificateurs: A, E

Logique Temporelle

connecteurs temporelles

Opérateur **X** "next"



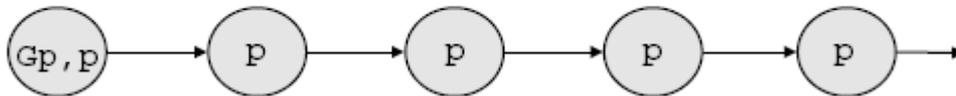
p serait vrai dans l'état suivant

Opérateur **F** "sometimes in the future"

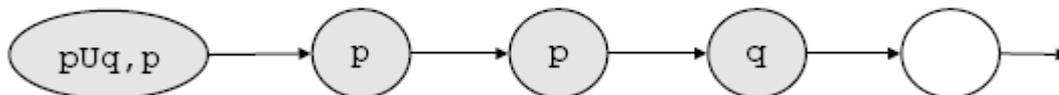


p serait vrai dans un état futur

Opérateur **G** "always in the future"



Opérateur **U** "p true until q true"



Remarque:

- **si φ_2 est vrai à un instant donc $\varphi_1 U \varphi_2$ l'est aussi**
- **Si $\varphi_1 U \varphi_2$ est vrai donc on doit avoir dans le futur φ_2 vrai aussi**
- **$\alpha U \beta = \beta \vee (\alpha \wedge X(\alpha U \beta)).$**

Logique Temporelle

exemples

Que signifie les formules suivantes :

- $F(x=0)$,
- $G\neg(x=0)$,
- $G(p \rightarrow Fq)$,
- $(\text{init} \wedge \text{précond}) \rightarrow F(\text{end} \wedge \text{poscond})$

Logique temporelle quantificateurs

- Les connecteurs temporelle expriment des propriétés sur **un seul chemin**, donc **un seul futur**
- Mais en réalité, on peut avoir plusieurs exécutions possibles: on introduit deux quantificateurs:

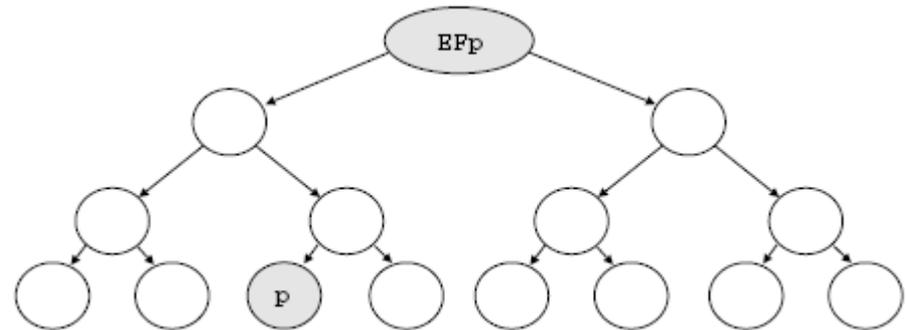
A: **tous les chemins** futurs **vérifient** la propriété

E: **certains chemins** futurs **vérifient** la propriété

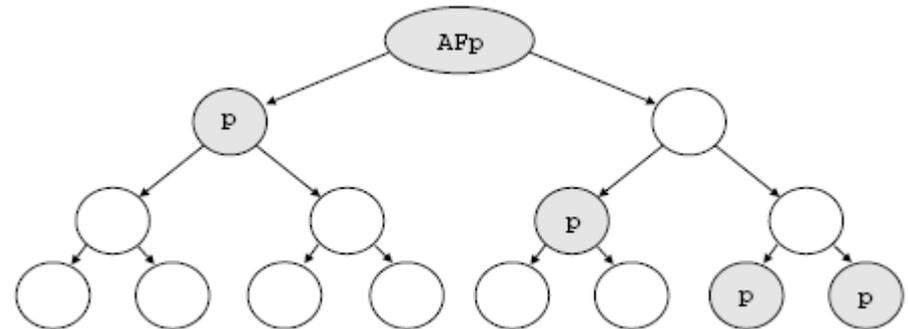
Logiques Temporelle

exemple 2

EFp : p vrai dans au moins un état



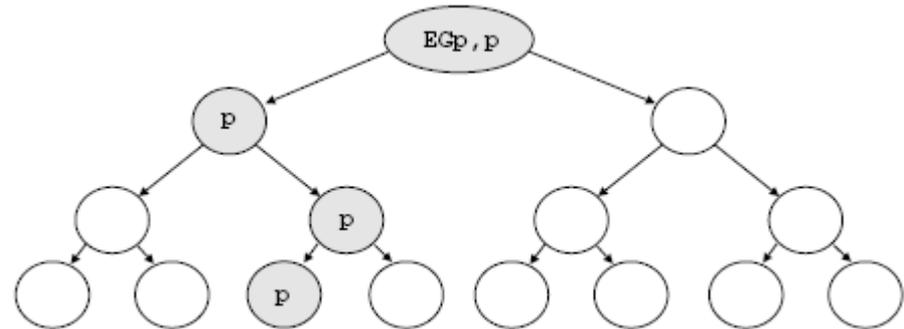
AFp : p atteignable par tous les chemins



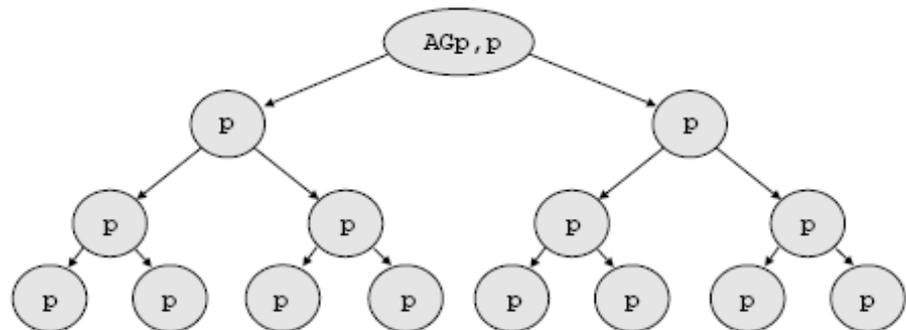
Logique temporelle

exemple 2

EG p : il existe un chemin avec p toujours vrai



AG p : p est toujours vrai



Test

- C'est quoi une logique? Pourquoi la logique?
- Pourquoi utiliser la logique temporelle?
- C'est quoi la nouveauté dans cette logique?
- C'est quoi le modèle de cette logique et montrer la sémantique de ses formules?
- Quelles sont les différentes classes de cette logique?

Logique LTL

- Une logique temporelle dite linéaire
- **Seulement A, mais pas de quantificateur E**

Exemple : $AFGp$ dans LTL
 EFp pas dans LTL

Logique LTL

sémantique sur un seul chemin

- Les formules de LTL :

$$\varphi ::= p \in AP \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \mathbf{X}\varphi \mid \mathbf{F}\varphi \mid \mathbf{G}\varphi \mid \varphi \mathbf{U}\varphi$$

- Le **modèle d'interprétation** des formules LTL est un **chemin d'exécution** : σ (suite d'états). $\sigma = e_0, e_1, e_2, \dots$
- On note par $(\sigma, i) \models \varphi$ le fait que φ est satisfaite dans l'état i appartenant au chemin σ

Logique LTL

sémantique sur un seul chemin

-La relation de satisfaction de formules LTL $\sigma \models \varphi$ est définie inductivement, comme suit :

- $\sigma \models p$ ssi p est satisfaite dans l'état e_0 et on note ça par $(\sigma, 0) \models p$
- $\sigma \models \varphi_1 \wedge \varphi_2$ ssi $(\sigma, 0) \models \varphi_1$ et $(\sigma, 0) \models \varphi_2$
- $\sigma \models \varphi_1 \vee \varphi_2$ ssi $(\sigma, 0) \models \varphi_1$ ou $(\sigma, 0) \models \varphi_2$
- $\sigma \models X\varphi$ ssi $(\sigma, 1) \models \varphi$
- $\sigma \models F\varphi$ ssi il existe un $k \geq 0$ tel que $(\sigma, k) \models \varphi$
- $\sigma \models G\varphi$ ssi pour tout $k \geq 0$ on a $(\sigma, k) \models \varphi$
- $\sigma \models \varphi_1 U \varphi_2$ ssi il existe $k \geq 0$ avec $(\sigma, k) \models \varphi_2$ et pour tout $j: 0 \leq j < k$:
 $(\sigma, j) \models \varphi_1$

Logique LTL

quelques équivalences

- $F\varphi \equiv \neg G\neg\varphi$
- $F\varphi \equiv \text{true}U\varphi$
- $\neg F\varphi \equiv G\neg\varphi$
- $\neg X\varphi \equiv X\neg\varphi$

?

LTL

sémantique sur une structure de Kripke

- Les formules concernent **un chemin** ou **plusieurs** qui partent d'un certain état s :

$$\varphi_p ::= p \in AP$$

$$| \neg\varphi_p | \varphi_p \vee \varphi_p | \varphi_p \wedge \varphi_p | \mathbf{X}\varphi_p | \mathbf{F}\varphi_p | \mathbf{G}\varphi_p | \varphi_p \mathbf{U}\varphi_p$$

$$\varphi_s ::= \mathbf{A}\varphi_p$$

- Le modèle est un couple (M, s) où M est une structure de Kripke et s un état dans cette structure

- Satisfaction

. $M, s \models_s \mathbf{A}\varphi_p$ ssi tous les chemins σ partant de s
vérifient $\sigma \models \varphi_p$

. $\sigma \models \varphi_p$ défini comme avant

- Enfinement $M \models \varphi$ ssi $M, s_0 \models_s \varphi$

LTL

exemples

$\neg G(r \wedge \neg o \wedge \neg v)$: le système ne reste pas tout le temps .

$G((\neg r \wedge o \wedge \neg v) \rightarrow X(r \wedge \neg o \wedge \neg v))$:  est tjs imm. suivi de .

$GF(\neg r \wedge \neg o \wedge v)$: le système passe infiniment souvent par .