SAPIENZA Università di Roma – MSc. in Engineering in Computer Science

Formal Methods

How to Write Formulas in Text

Sets

Syntax in math:

$$S := \{a, b, c\} \mid S_1 \cap S_2 \mid S_1 \cup S_2 \mid S_1 - S_2 \mid S_1 \times S_2$$
$$S_1 \subseteq S_2 \mid S_1 = S_2 \mid a \in S$$

Syntax in text:

```
S := {a,b,c} | S1 inter S2 | S1 union S2 | S1 - S2 | S1 X S2
               S1 subseteq S2 | S1 = S2 | a in S
```

Example:

$(A \cup B) \subseteq C$ `(A union B) subseteq C`

Propositional Logic

Syntax in math:

$$\varphi := A \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2 \mid \varphi_1 \supset \varphi_2 \mid \varphi_1 \equiv \varphi_2$$

Syntax in text:

```
P := A | not P | P1 and P2 | P1 or P2 | P1 implies P2 | P1 equiv P2
```

Example:

$(A \vee \neg B) \supset C$ `(A or not B) implies C`

First Order Logic

Syntax in math:

$$\varphi := A(t_1, \ldots, t_n) \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2 \mid \varphi_1 \supset \varphi_2 \mid \varphi_1 \equiv \varphi_2 \mid \exists x.\varphi \mid \forall x.\varphi$$

Syntax in text:

```
P := A(t1,...,t2) | not P | P1 and P2 | P1 or P2 | P1 implies P2 | P1 equiv P2 |
     exists x. P | forall x. P
```

Example:

$\forall x.Student(x) \supset Person(x)$ `forall x. Student(x) implies Person(x)`

Mu-Calculus

Syntax in math:

$$\varphi := A \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2 \mid \varphi_1 \supset \varphi_2 \mid \varphi_1 \equiv \varphi_2 \mid \mu X.\varphi \mid \nu X.\varphi \mid X$$

Syntax in text:

```
P := A | not P | P1 and P2 | P1 or P2 | P1 implies P2 | P1 equiv P2
     mu X.P | nu X.P | X
```

Example:

$\mu X.\nu Y.((a \wedge \langle next\rangle X) \vee (b \wedge [next]Y))$ `mu X. nu Y. ((a and <next>X) or (b and [next]Y))`

CTL

Syntax in math:

$$\begin{aligned}\varphi := \;& A \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2 \mid \varphi_1 \supset \varphi_2 \mid \varphi_1 \equiv \varphi_2 \mid \\ & AX\varphi \mid AF\varphi \mid AG\varphi \mid \varphi_1 AU \varphi_2 \mid \\ & EX\varphi \mid EF\varphi \mid EG\varphi \mid \varphi_1 EU \varphi_2\end{aligned}$$

Syntax in text:

```
P := A | not P | P1 and P2 | P1 or P2 | P1 implies P2 | P1 equiv P2
     AX P | AF P | AG P | P1 AU P2 |
     EX P | EF P | EG P | P1 EU P2
```

Example:

$AF(a \wedge AX(EGb))$ `AF(a and AX (EG b))`

LTL

Syntax in math:

$$\begin{aligned}\varphi := \;& A \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2 \mid \varphi_1 \supset \varphi_2 \mid \varphi_1 \equiv \varphi_2 \mid \\ & \bigcirc\varphi \mid \Diamond\varphi \mid \Box\varphi \mid \varphi_1 U \varphi_2\end{aligned}$$

Syntax in text:

```
P := A | not P | P1 and P2 | P1 or P2 | P1 implies P2 | P1 equiv P2
     next P | eventually P | always P | P1 until P2
```

Example:

$\Box(a \supset \bigcirc\Diamond b)$ `always (a implies next eventually b)`