

# PBES Generation

Wieger Wesselink, Tim Willemse

April 26, 2024

This document describes the generation of PBESs. These algorithms are implemented in the tools `lps2pbes` and `lpsbisim2pbes`.

## 1 Transforming an LPS and a property to a PBES

In this section we define the algorithm `LPS2PBES` that generates a PBES from a modal mu calculus property  $\varphi$  and an LPS. Let  $\langle D_p, d_0, P \rangle$  be the LPS given by

$$\begin{aligned} \mathbf{proc} \ P(x:D_p) &= \sum_{i \in I} \sum_{y_i: E_i} c_i(x, y_i) \rightarrow a_i(f_i(x, y_i))^{t_i}(x, y_i) \cdot P(g_i(x, y_i)) \\ &+ \sum_{j \in J} \sum_{y_j: E_j} c_j(x, y_j) \rightarrow \delta^{t_j}(x, y_j); \end{aligned}$$

where  $a_i(f_i(x, y))$  is a multiset of actions. Then we define

$$\mathbf{LPS2PBES}(\sigma X(x_f : D_f := d). \varphi, \langle D_p, d_0, P \rangle) = \mathbf{E}(\varphi),$$

where the function  $\mathbf{E}$  is inductively defined using the tables below. The function  $\varphi$  has to be in positive normal form, i.e. it may not contain any  $\neg$  or  $\Rightarrow$  symbols. There is also an untimed variant of the algorithm, which can be obtained by removing all time references. A formula  $\varphi$  not of the form  $\sigma X(x_f : D_f := d). \varphi$  is first translated into  $\nu X(). \varphi$ . We assume that  $T : \mathbb{R}$  is a unique fresh time variable that is generated by the algorithm. We denote the formula  $\varphi$  that is passed to the algorithm `LPS2PBES` as  $\varphi_0$ . It is used as a global variable in the translation.

Let  $a = \{a_1, \dots, a_n\}$  and  $b = \{b_1, \dots, b_n\}$  be two multi actions. Let  $A$  be the set of all permutations  $[i_1, \dots, i_n]$  of  $[1, \dots, n]$  such that  $\text{name}(a_k) = \text{name}(b_{i_k})$  for  $k = 1 \dots n$ . Then we define the function  $\mathbf{Sat}$  as follows:

$$\begin{aligned} \mathbf{Sat}(a^t, b) &=_{def} \begin{cases} \bigvee_{[i_1, \dots, i_n] \in A} \bigwedge_{k=1 \dots n} (a_k = b_{i_k}) & \text{if } A \neq \emptyset \\ false & \text{otherwise} \end{cases} \\ \mathbf{Sat}(a^t, c) &=_{def} c \\ \mathbf{Sat}(a^t, \alpha^c u) &=_{def} \mathbf{Sat}(a^t, \alpha) \wedge t \approx u \\ \mathbf{Sat}(a^t, \neg \alpha) &=_{def} \neg \mathbf{Sat}(a^t, \alpha) \\ \mathbf{Sat}(a^t, \alpha \wedge \beta) &=_{def} \mathbf{Sat}(a^t, \alpha) \wedge \mathbf{Sat}(a^t, \beta) \\ \mathbf{Sat}(a^t, \alpha \vee \beta) &=_{def} \mathbf{Sat}(a^t, \alpha) \vee \mathbf{Sat}(a^t, \beta) \\ \mathbf{Sat}(a^t, \alpha \Rightarrow \beta) &=_{def} \mathbf{Sat}(a^t, \alpha) \Rightarrow \mathbf{Sat}(a^t, \beta) \\ \mathbf{Sat}(a^t, \forall x:D. \alpha) &=_{def} \forall y:D. (\mathbf{Sat}(a^t, \alpha[x := y])) \\ \mathbf{Sat}(a^t, \exists x:D. \alpha) &=_{def} \exists y:D. (\mathbf{Sat}(a^t, \alpha[x := y])) \end{aligned}$$

$$\begin{array}{ll}
\mathbf{Par}_{X,l}(c) & =_{def} \square \\
\mathbf{Par}_{X,l}(\neg\varphi) & =_{def} \mathbf{Par}_{X,l}(\varphi) \\
\mathbf{Par}_{X,l}(\varphi \wedge \psi) & =_{def} \mathbf{Par}_{X,l}(\varphi) + +\mathbf{Par}_{X,l}(\psi) \\
\mathbf{Par}_{X,l}(\varphi \vee \psi) & =_{def} \mathbf{Par}_{X,l}(\varphi) + +\mathbf{Par}_{X,l}(\psi) \\
\mathbf{Par}_{X,l}(\varphi \Rightarrow \psi) & =_{def} \mathbf{Par}_{X,l}(\varphi) + +\mathbf{Par}_{X,l}(\psi) \\
\mathbf{Par}_{X,l}([\alpha]\varphi) & =_{def} \mathbf{Par}_{X,l}(\varphi) \\
\mathbf{Par}_{X,l}(\langle\alpha\rangle\varphi) & =_{def} \mathbf{Par}_{X,l}(\varphi) \\
\mathbf{Par}_{X,l}(\forall x:D.\varphi) & =_{def} \mathbf{Par}_{X,l++[x:D]}(\varphi) \\
\mathbf{Par}_{X,l}(\exists x:D.\varphi) & =_{def} \mathbf{Par}_{X,l++[x:D]}(\varphi) \\
\mathbf{Par}_{X,l}(Y(d_f)) & =_{def} \square \\
\mathbf{Par}_{X,l}(\sigma Y(x_f:D_f := d).\varphi) & =_{def} \begin{cases} l & \text{if } Y = X \\ \mathbf{Par}_{X,l++[x_f:D_f]}(\varphi) & \text{if } Y \neq X \end{cases} \\
\mathbf{Par}_{X,l}(\nabla(t)) & =_{def} \square \\
\mathbf{Par}_{X,l}(\Delta(t)) & =_{def} \square
\end{array}$$

$$\begin{array}{lll}
\mathbf{RHS}(c) & =_{def} & c \\
\mathbf{RHS}(\varphi \wedge \psi) & =_{def} & \mathbf{RHS}(\varphi) \wedge \mathbf{RHS}(\psi) \\
\mathbf{RHS}(\varphi \vee \psi) & =_{def} & \mathbf{RHS}(\varphi) \vee \mathbf{RHS}(\psi) \\
\mathbf{RHS}(\forall x:D.\varphi) & =_{def} & \forall x:D.\mathbf{RHS}(\varphi) \\
\mathbf{RHS}(\exists x:D.\varphi) & =_{def} & \exists x:D.\mathbf{RHS}(\varphi) \\
\mathbf{RHS}([\alpha]\varphi) & =_{def} & \bigwedge_{i \in I} \forall y:E_i ((\mathbf{Sat}(a_i(f_i(x, y))) \text{c} t_i(x, y), \alpha) \wedge \\
& & c_i(x, y) \wedge t_i(x, y) > T) \Rightarrow \\
& & \mathbf{RHS}(\varphi)[T, x := t_i(x, y), g_i(x, y)]) \\
\mathbf{RHS}(\langle \alpha \rangle \varphi) & =_{def} & \bigvee_{i \in I} \exists y:E_i ((\mathbf{Sat}(a_i(f_i(x, y))) \text{c} t_i(x, y), \alpha) \wedge \\
& & c_i(x, y) \wedge t_i(x, y) > T \wedge \\
& & \mathbf{RHS}(\varphi)[T, x := t_i(x, y), g_i(x, y)]) \\
\mathbf{RHS}(X(e)) & =_{def} & X(T, e, x, \mathbf{Par}_{X, \square}(\varphi_0)) \\
\mathbf{RHS}(\sigma X(x_f:D_f := e). \varphi) & =_{def} & X(T, e, x, \mathbf{Par}_{X, \square}(\varphi_0)) \\
\mathbf{RHS}(\nabla(t)) & =_{def} & (\bigwedge_{i \in I \cup J} \forall y:E_i ((\neg c_i(x, y) \vee t > t_i(x, y))) \wedge t > T \\
\mathbf{RHS}(\Delta(t)) & =_{def} & (\bigvee_{i \in I \cup J} \exists y:E_i ((c_i(x, y) \wedge t \leq t_i(x, y))) \vee t \leq T
\end{array}$$

$$\begin{array}{lll}
\mathbf{E}(c) & =_{def} & \epsilon \\
\mathbf{E}(\varphi \wedge \psi) & =_{def} & \mathbf{E}(\varphi)\mathbf{E}(\psi) \\
\mathbf{E}(\varphi \vee \psi) & =_{def} & \mathbf{E}(\varphi)\mathbf{E}(\psi) \\
\mathbf{E}(\forall x:D.\varphi) & =_{def} & \mathbf{E}(\varphi) \\
\mathbf{E}(\exists x:D.\varphi) & =_{def} & \mathbf{E}(\varphi) \\
\mathbf{E}([\alpha]\varphi) & =_{def} & \mathbf{E}(\varphi) \\
\mathbf{E}(\langle \alpha \rangle \varphi) & =_{def} & \mathbf{E}(\varphi) \\
\mathbf{E}(\nabla) & =_{def} & \epsilon \\
\mathbf{E}(\nabla(t)) & =_{def} & \epsilon \\
\mathbf{E}(\Delta) & =_{def} & \epsilon \\
\mathbf{E}(\Delta(t)) & =_{def} & \epsilon \\
\mathbf{E}(X(e)) & =_{def} & \epsilon \\
\mathbf{E}(\sigma X(x_f:D_f := e). \varphi) & =_{def} & (\sigma X(T : \mathbb{R}, x_f:D_f, x:D_p, \mathbf{Par}_{X, \square}(\varphi_0)) = \mathbf{RHS}(\varphi) ) \mathbf{E}(\varphi)
\end{array}$$

## 1.1 Counter example generation

There is a modified translation that adds information to the PBES from which a counter example can be generated. N.B. The counter example generation is only available for the untimed case. The function  $\mathbf{RHS}$  is adapted from

$$\begin{array}{ll}
\mathbf{RHS}([\alpha]\varphi) & =_{def} \bigwedge_{i \in I} \forall y:E_i ((\mathbf{Sat}(a_i(f_i(x, y))), \alpha) \wedge c_i(x, y)) \Rightarrow \mathbf{RHS}(\varphi)[x := g_i(x, y)] \\
\mathbf{RHS}(\langle \alpha \rangle \varphi) & =_{def} \bigvee_{i \in I} \exists y:E_i ((\mathbf{Sat}(a_i(f_i(x, y))), \alpha) \wedge c_i(x, y)) \wedge \mathbf{RHS}(\varphi)[x := g_i(x, y)]
\end{array}$$

into

$$\begin{array}{ll}
\mathbf{RHS}([\alpha]\varphi) & =_{def} \bigwedge_{i \in I} \forall y:E_i (\mathbf{Sat}(a_i(f_i(x, y))), \alpha) \wedge c_i(x, y) \Rightarrow \\
& ((\mathbf{RHS}(\varphi)[x := g_i(x, y)] \wedge Z_{a_i}^+(x, f_i(x, y), g_i(x, y))) \vee Z_{a_i}^-(x, f_i(x, y), g_i(x, y))) \\
\mathbf{RHS}(\langle \alpha \rangle \varphi) & =_{def} \bigvee_{i \in I} \exists y:E_i (\mathbf{Sat}(a_i(f_i(x, y))), \alpha) \wedge c_i(x, y) \wedge \\
& ((\mathbf{RHS}(\varphi)[x := g_i(x, y)] \vee Z_{a_i}^-(x, f_i(x, y), g_i(x, y))) \wedge Z_{a_i}^+(x, f_i(x, y), g_i(x, y))),
\end{array}$$

where  $\nu Z_{a_i}^+(x, f_i(x, y), d') = true$  and  $\nu Z_{a_i}^-(x, f_i(x, y), d') = false$  are additional equations for every  $a_i$  that appears in the LPS.

## 2 Transforming an LTS and a property to a PBES

In this section we define the algorithm `LTS2PBES` that generates a PBES from a modal mu calculus formula  $\varphi$  and an LTS  $\langle S, Act, \rightarrow, s_0 \rangle$ :

$$\text{LTS2PBES}(\sigma X(d : D := e). \varphi, \langle S, Act, \rightarrow, s_0 \rangle) = \langle \mathbf{E}_{\sigma X(d:D:=e). \varphi}(\varphi), X_{s_0}(e) \rangle$$

where the function  $\mathbf{E}$  is inductively defined using the tables below. The function  $\varphi$  has to be in positive normal form, i.e. it may not contain any  $\neg$  or  $\Rightarrow$  symbols. A formula  $\psi$  not of the form  $\sigma X(d : D := e). \varphi$  is translated into  $\nu X(). \psi$ . We denote the formula  $\varphi$  that is passed to the algorithm `LTS2PBES` as  $\varphi_0$ . It is used as a global variable in the translation.

N.B. The functions **Sat** and **Par** are defined in the section about `LPS2PBES`.

$\mathbf{E}(c)$	$=_{def}$	$\epsilon$
$\mathbf{E}(\varphi \wedge \psi)$	$=_{def}$	$\mathbf{E}(\varphi)\mathbf{E}(\psi)$
$\mathbf{E}(\varphi \vee \psi)$	$=_{def}$	$\mathbf{E}(\varphi)\mathbf{E}(\psi)$
$\mathbf{E}(\forall x:D.\varphi)$	$=_{def}$	$\mathbf{E}(\varphi)$
$\mathbf{E}(\exists x:D.\varphi)$	$=_{def}$	$\mathbf{E}(\varphi)$
$\mathbf{E}([\alpha]\varphi)$	$=_{def}$	$\mathbf{E}(\varphi)$
$\mathbf{E}(\langle \alpha \rangle \varphi)$	$=_{def}$	$\mathbf{E}(\varphi)$
$\mathbf{E}(X(d))$	$=_{def}$	$\epsilon$
$\mathbf{E}(\sigma X(d:D := e). \varphi)$	$=_{def}$	$(\sigma\{X_s(d : D, \mathbf{Par}_{X, \square}(\varphi_0)) = \mathbf{RHS}_\varphi(\varphi, s) \mid s \in S\}) \mathbf{E}(\varphi)$

$\mathbf{RHS}(c, s)$	$=_{def}$	$c$
$\mathbf{RHS}(\varphi \wedge \psi, s)$	$=_{def}$	$\mathbf{RHS}(\varphi, s) \wedge \mathbf{RHS}(\psi, s)$
$\mathbf{RHS}(\varphi \vee \psi, s)$	$=_{def}$	$\mathbf{RHS}(\varphi, s) \vee \mathbf{RHS}(\psi, s)$
$\mathbf{RHS}(\forall d:D.\varphi, s)$	$=_{def}$	$\forall d:D.\mathbf{RHS}(\varphi, s)$
$\mathbf{RHS}(\exists d:D.\varphi, s)$	$=_{def}$	$\exists d:D.\mathbf{RHS}(\varphi, s)$
$\mathbf{RHS}([\alpha]\varphi, s)$	$=_{def}$	$\bigwedge \left\{ \mathbf{Sat}(a(x), \alpha) \Rightarrow \mathbf{RHS}_{\varphi_0}(\varphi, t) \mid s \xrightarrow{a(x)} t \right\}$
$\mathbf{RHS}(\langle \alpha \rangle \varphi, s)$	$=_{def}$	$\bigvee \left\{ \mathbf{Sat}(a(x), \alpha) \wedge \mathbf{RHS}_{\varphi_0}(\varphi, t) \mid s \xrightarrow{a(x)} t \right\}$
$\mathbf{RHS}(X(e), s)$	$=_{def}$	$X_s(e, \mathbf{Par}_{X, \square}(\varphi_0))$
$\mathbf{RHS}(\sigma X(d:D := e). \varphi, s)$	$=_{def}$	$X_s(e, \mathbf{Par}_{X, \square}(\varphi_0))$

## 2.1 Counter example generation

There is a modified translation that adds information to the PBES from which a counter example can be generated. The function RHS is adapted from

$$\begin{aligned} \mathbf{RHS}([\alpha]\varphi, s) &=_{def} \bigwedge \left\{ \mathbf{Sat}(a(x), \alpha) \Rightarrow \mathbf{RHS}_{\varphi_0}(\varphi, t) \mid s \xrightarrow{a(x)} t \right\} \\ \mathbf{RHS}(\langle\alpha\rangle\varphi, s) &=_{def} \bigvee \left\{ \mathbf{Sat}(a(x), \alpha) \wedge \mathbf{RHS}_{\varphi_0}(\varphi, t) \mid s \xrightarrow{a(x)} t \right\} \end{aligned}$$

into

$$\begin{aligned} \mathbf{RHS}([\alpha]\varphi, s) &=_{def} \bigwedge \left\{ \mathbf{Sat}(a(x), \alpha) \Rightarrow ((\mathbf{RHS}(\varphi, t) \wedge Z_a^+(s, x, t)) \vee Z_a^-(s, x, t)) \mid s \xrightarrow{a(x)} t \right\} \\ \mathbf{RHS}(\langle\alpha\rangle\varphi, s) &=_{def} \bigvee \left\{ \mathbf{Sat}(a(x), \alpha) \wedge ((\mathbf{RHS}(\varphi, t) \vee Z_a^-(s, x, t)) \wedge Z_a^+(s, x, t)) \mid s \xrightarrow{a(x)} t \right\} \end{aligned}$$

where  $\nu Z_{a_i}^+(s, d_x, t) = true$  and  $\nu Z_{a_i}^-(s, d_x, t) = false$  are additional equations for every  $a(x)$  that appears in the LTS.

### 3 Bisimulation

Let

$$\begin{aligned} M(d) &= \sum_{i \in I_M} \sum_{e: E_i} c_i(d, e) \rightarrow a_i(d, e) \cdot M(g_i(d, e)) \\ S(d) &= \sum_{i \in I_S} \sum_{e: E_i} c_i(d, e) \rightarrow a_i(d, e) \cdot M(g_i(d, e)) \end{aligned}$$

be two linear processes, such that  $I_M \cap I_S = \emptyset$ .  $M$  is called the model and  $S$  the specification. The expression  $a_i(d, e)$  can be a multi-action, or have the special value  $\tau$ . We assume that there are no  $\delta$  summands. We define four pbes equation systems that express some kind of bisimulation equivalence between  $M$  and  $S$ .

**Branching Bisimulation**  $brbsim(M, S) = \nu E_2 \mu E_1$ , where

$$\begin{aligned} E_2 &:= \{X^{M,S}(d, d') = match^{M,S}(d, d') \wedge match^{S,M}(d', d), \\ &\quad X^{S,M}(d', d) = X^{M,S}(d, d')\} \\ E_1 &:= \{Y_i^{M,S}(d, d', e) = close_i^{M,S}(d, d', e) | i \in I_M, \\ &\quad Y_i^{S,M}(d', d, e) = close_i^{S,M}(d', d, e) | i \in I_S\} \end{aligned}$$

with for all  $i \in I_p$  and  $(p, q) \in \{(M, S), (S, M)\}$ :

$$\begin{aligned} match^{p,q}(d, d') &= \bigwedge_{i \in I_p} \forall e: E_i. (c_i(d, e) \Rightarrow Y_i^{p,q}(d, d', e)) \\ close_i^{p,q}(d, d', e) &= \bigvee_{\{j \in I_q | a_j = \tau\}} \exists e': E_j. (c_j(d', e') \wedge Y_i^{p,q}(d, g_j(d', e'), e)) \\ &\quad \vee (X^{p,q}(d, d') \wedge step_i^{p,q}(d, d', e)) \\ step_i^{p,q}(d, d', e) &= \begin{cases} a_i = \tau : & X^{p,q}(g_i(d, e), d') \vee \bigvee_{\{j \in I_q | a_j = \tau\}} \exists e': E_j. (c_j(d', e') \wedge X^{p,q}(g_i(d, e), g_j(d', e'))) \\ a_i \neq \tau : & \bigvee_{\{j \in I_q | a_j = a_i\}} \exists e': E_j. (c_j(d', e') \wedge (a_i(d, e) = a_j(d', e')) \wedge X^{p,q}(g_i(d, e), g_j(d', e'))) \end{cases} \end{aligned}$$

**Strong Bisimulation**  $sbsim(M, S) = \nu E$ , where

$$E := \{X^{M,S}(d, d') = match^{M,S}(d, d') \wedge match^{S,M}(d', d), \\ X^{S,M}(d', d) = X^{M,S}(d, d')\}$$

with for all  $i \in I_p$  and  $(p, q) \in \{(M, S), (S, M)\}$ :

$$\begin{aligned} match^{p,q}(d, d') &= \bigwedge_{i \in I_p} \forall e: E_i. (c_i(d, e) \Rightarrow step_i^{p,q}(d, d', e)) \\ step_i^{p,q}(d, d', e) &= \bigvee_{j \in I_q} \exists e': E_j. (c_j(d', e') \wedge (a_i(d, e) = a_j(d', e')) \wedge X^{p,q}(g_i(d, e), g_j(d', e'))) \end{aligned}$$

**Weak Bisimulation**  $wbsim(M, S) = \nu E_2 \mu E_1$ , where

$$\begin{aligned} E_3 &:= \{X^{M,S}(d, d') = match^{M,S}(d, d') \wedge match^{S,M}(d', d), \\ &\quad X^{S,M}(d', d) = X^{M,S}(d, d')\} \\ E_2 &:= \{Y_{1,i}^{M,S}(d, d', e) = close_{1,i}^{M,S}(d, d', e) | i \in I_M, \\ &\quad Y_{2,i}^{M,S}(d, d') = close_{2,i}^{M,S}(d, d') | i \in I_M, \\ &\quad Y_{1,i}^{S,M}(d', d, e) = close_{1,i}^{S,M}(d', d, e) | i \in I_S, \\ &\quad Y_{2,i}^{S,M}(d', d) = close_{2,i}^{S,M}(d', d) | i \in I_S\} \end{aligned}$$

with for all  $i \in I_p$  and  $(p, q) \in \{(M, S), (S, M)\}$ :

$$\begin{aligned}
match^{p,q}(d, d') &= \bigwedge_{i \in I_p} \forall e: E_i. (c_i(d, e) \Rightarrow Y_{1,i}^{p,q}(d, d', e)) \\
close_{1,i}^{p,q}(d, d', e) &= \left( \bigvee_{\{j \in I_q \mid a_j = \tau\}} \exists e': E_j. (c_j(d', e') \wedge Y_{1,i}^{p,q}(d, g_j(d', e'), e)) \right) \vee step_i^{p,q}(d, d', e) \\
step_i^{p,q}(d, d', e) &= \begin{cases} a_i = \tau : close_{2,i}^{p,q}(g_i(d, e), d') \\ a_i \neq \tau : \bigvee_{j \in I_q} \exists e': E_j. (c_j(d', e') \wedge a_i(d, e) = a_j(d', e') \wedge close_{2,i}^{p,q}(g_i(d, e), g_j(d', e'))) \end{cases} \\
close_{2,i}^{p,q}(d, d') &= X^{p,q}(d, d') \vee \bigvee_{\{j \in I_q \mid a_j = \tau\}} (\exists e': E_j. c_j(d', e') \wedge Y_{2,i}^{p,q}(d, g_j(d', e')))
\end{aligned}$$

**Branching Simulation Equivalence**  $brbsim(M, S) = \nu E_2 \mu E_1$ , where

$$\begin{aligned}
E_2 &:= \{X^{M,S}(d, d') = match^{M,S}(d, d') \wedge match^{S,M}(d', d), \\
&\quad X^{M,S}(d, d') = X^{S,M}(d', d), \\
&\quad X^{S,M}(d', d) = X^{M,S}(d, d')\} \\
E_1 &:= \{Y_i^{M,S}(d, d', e) = close_i^{M,S}(d, d', e) \mid i \in I_M, \\
&\quad Y_i^{S,M}(d', d, e) = close_i^{S,M}(d', d, e) \mid i \in I_S\}
\end{aligned}$$

with  $match$ ,  $close$ , and  $step$  defined exactly the same as in branching bisimulation.