Alternative encoding for learning feature-based reformulations

Version 22/07/2018

We aim at a reduction to SAT with parameters K and N: K bounds the size of the abstract space by selecting K features from the set of available features, and N bounds the number of abstract actions that can be used in the abstract space.

The general idea is to use "meta-variables" for selecting the features that define the abstract state model and then using "meta-variables" that define the abstact actions over the abstract states. The SAT problem "constructs" the abstract model and then verifies soundness and completeness with respect to the transitions and states in the sample.

The set of available features is denoted with $\mathcal{F}$ and the set of sampled transitions as $\mathcal{M}$. The value of features in $\mathcal{F}$ at the states in the sample $\mathcal{M}$ are given as input to the reduction in the form of a matrix M that has as many rows as the number of features in $\mathcal{F}$ and as many columns as the number of states in $\mathcal{M}$. We denote with S and T the set of states and transitions in $\mathcal{M}$.

If the unique states in $\mathcal{M}$ are $\{s_0, s_1, \ldots\}$, we use integer i to denote state s_i. For boolean features f and state s_i, the entry m_{fi} in the matrix M is 1 iff the feature f holds at state s_i. For numerical features f and state s_i, the entry m_{fi} is the value of f at state i.

The size of the encoding is summarized in the table below where S is number of states, T is number of transitions, F is number of features, $L = \log_2(F)$, $\Delta \leq L$, D and U are numbers for dummy and inexistent features respectively, and $Q(L, K, N) = N(1 + 4K) + K(D + U) + (K - 1)(L^2 + L) - 1$.

Parameters:	$K = \#$ selected features, $N = \#$ abstract actions
# variables:	$2SK(1 + N) + T(2K + 3N) + 2K(2N + L) - L$
# clauses:	$S(2KF + 7KN + N) + T(2KF + NK(10 + \Delta) + 2N + 1) + Q(L, K, N)$

The next table shows some experimental data over Blocksworld and Gripper:

				Input			SAT Theory		
problem	status	K	N	#states	#trans.	#features	#vars	#clauses	Minisat
Blocks04	SAT	5	8	125	272	73	20,721	472,560	1.63
Blocks04	UNSAT	5	7	125	272	73	18,635	449,815	>250
Blocks05	SAT	5	8	866	2,090	102	149,223	4,387,384	16.31
Blocks05	UNSAT	5	7	866	2,090	102	134,273	4,216,157	>250
Gripper03	SAT	5	10	88	280	12	21,116	264,029	0.28
Gripper03	UNSAT	5	9	88	280	12	19,376	242,080	1.43
Gripper04	SAT	5	10	256	896	14	64,236	854,955	5.00
Gripper04	UNSAT	5	9	256	896	14	58,968	785,686	6.86
Gripper05	SAT	5	10	704	2,624	16	182,636	2,546,913	2.47
Gripper05	UNSAT	5	9	704	2,624	16	167,704	2,345,740	39.16
Gripper06	SAT	5	10	1,856	7,232	18	493,685	7,518,671	39.25
Gripper06	UNSAT	5	9	1,856	7,232	18	453,409	6,931,130	110.45

1 Preliminaries

The sample consists of transitions of the form $t = \langle s, a, s' \rangle$ that are assumed to come from one (**** CHECK: or more than one? ****) instances of a generalized problem.

The K features selected from $\mathcal{F}$ define an abstract space given by the 2^K boolean valuations of the K features. For boolean features, such a valuation contains the value of the feature. For numerical features,

the boolean value denotes whether the numerical feature is equal to or bigger than zero. An abstract state represents a set of concrete states in the sample. Abstract actions look like a regular STRIPS action on the abstract space. An abstract action $\tilde{a}$ is made of a precondition and an effect, both being partial valuations of the K features. For boolean features f, f (resp. $\neg f$) appears in the precondition iff $\tilde{a}$ requires f to hold (resp. not hold) in the abstract state, and f (resp. $\neg f$) appears in the effect if $\tilde{a}$ makes f true (resp. false). For numerical effects, positive (resp. negative) f-literals in the precondition stand for the condition $f > 0$ (resp. $f = 0$), and positive (resp. negative) f-literals in the effect stand for an increment (resp. decrement) of f. Hence, whether the feature f is boolean or numeric, we can specify the precondition or effect of an abstract action on f with a boolean f-literal.

Once abstract actions are defined over the abstract space, verifying soundness is just a matter of checking whether for each abstract transition, there is a matching concrete transition. That is, let us suppose that the abstract state $\tilde{s}$ is mapped by the abstract action $\tilde{a}$ into the abstract state $\tilde{s}'$. The abstract action $\tilde{a}$ is made of a precondition and effect. The precondition must hold at $\tilde{s}$ while the effect maps $\tilde{s}$ into $\tilde{s}'$. We say that $\tilde{a}$ is sound (wrt $\mathcal{M}$) iff for each concrete state s on which the precondition of $\tilde{a}$ holds, there is transition $t = \langle s, a, s' \rangle$ in $\mathcal{M}$ that "matches" the abstract transition $\tilde{t} = \langle \tilde{s}, \tilde{a}, \tilde{s}' \rangle$. That is, s is represented by $\tilde{s}$ (i.e. for each of the K features f, the boolean value of f in s' and $\tilde{s}'$ coincide), s' is represented by $\tilde{s}'$, and the selected numerical features change in the transition t as indicated by the abstact action $\tilde{a}$: if f is increased (resp. decreased) by $\tilde{a}$, then f increases (resp. decreases) along the transition t.

Completeness is the opposite. The abstract model is complete if for each transition $t = \langle s, a, s' \rangle$ there is an abstract transition $\tilde{t} = \langle \tilde{s}, \tilde{a}, \tilde{s}' \rangle$ that matches t.

Let $\sim$ be the equivalance relation over concrete states defined by the K selected features: $s \sim s'$ iff for each selected feature f, the boolean value of f at s and s' are the same. For state s and abstract state $\tilde{s}$, we write $s \sim \tilde{s}$ when the boolean value of the K selected features coincide in s and $\tilde{s}$. Likewise, we denote with $[s]$ and $[\tilde{s}]$ the equivalence class for state s and the set $\{s : s \sim \tilde{s}\}$ respectively, and we write $t \sim \tilde{t}$, for transition t and abstract transition $\tilde{t}$, when t matches $\tilde{t}$.

2 Encoding

The input to the reduction consists of the matrix M and the parameters K and N. We assume that the set of feature $\mathcal{F}$ is enumerated as $f_0, f_1, \ldots$ in such a way that the boolean features appear only after the numerical features, and that the first boolean feature f_i has index i that is a power of 2. The gap between the last numerical feature and the first boolean feature is assumed to be filled with dummy features. With this encoding, checking whether a selected feature f_i is numerical just amounts to check whether its index is less than the index of the first boolean feature.

For each $j \in K$, let F^j be a multi-valued (meta-)variable with domain F that denotes the j-th feature selected from $\mathcal{F}$. Each F^j variable is encoded with $\lceil \log_2 F \rceil$ propositional variables F^j_k where F is the number of features in $\mathcal{F}$. To ensure different features are selected, it is enough that $F^i < F^j$ for $i < j$.

In the construction of the abstract model, we only consider abstract actions that put a precondition on the features affected by the action; i.e., an abstract action cannot affect a feature for which there is no precondition, but it may put a precondition on an unaffected feature.

Given K selected features, there are at most 9^K different abstract actions. Observe that each abstract action is the result of selecting a subset of affected features from the K features and considering the ways in which they can be affected (2 ways for each feature), and selecting a subset of features for preconditions and considering the ways the can appear as such (2 ways for each feature).[1] Hence, a bound on the total number of abstract actions is

$$\sum_{k=1}^{K} \binom{K}{k} 2^k \times \sum_{k=0}^{K} \binom{K}{k} 2^k \;=\; 3^K \times (3^K - 1) \;\leq\; 9^K \,. \tag{1}$$

Since K is constant, this number is also constant.

Let A^j, for $j \in \{0, \ldots, N-1\}$, denote the j-th abstract action. Instead of using an exact enumeration on the different abstract actions, we consider the following loose enumeration. As mentioned above, each

[1] The bound is not tight because we will require $f > 0$ in precondition of actions that decrement f.

abstract action is the result of selecting a subset of the selected features as affected, putting an effect over such features, and then adding preconditions. Thus, each abstract action can be specified with K bits to selected the subset of affected features, K bits for speciyfing the effect on each affected feature, K additional bits that select preconditions, and a final block of K bits that specify the value of the preconditions. In total, we need $4K$ bits to specify an abstract action. We use propositional variables $A^j_{i,k}$ for $j \in N$, $i \in \{1, \ldots, 4\}$, and $k \in \{0, \ldots, K-1\}$ to specify the abstract action A^j where

- the group $A^j_{1,k}$ selects the affected features,
- the group $A^j_{2,k}$ sets the effect for the selected features,
- the group $A^j_{3,k}$ selects the features for preconditions, and
- the group $A^j_{4,k}$ sets the precondition for the additional features.

We assume that the precondition $f > 0$ must be present when f is decremented, and we require that each abstract action affects at least one feature. These restrictions are enforced with:

$$A^j_{2,k} \rightarrow A^j_{1,k}\,, \tag{2}$$

$$A^j_{4,k} \rightarrow A^j_{3,k}\,, \tag{3}$$

$$A^j_{1,k} \wedge \neg A^j_{2,k} \wedge [F^k \text{ is numeric}] \rightarrow A^j_{4,k}\,, \tag{4}$$

$$A^j_{1,0} \vee A^j_{1,1} \vee \cdots \vee A^j_{1,K-1}\,. \tag{5}$$

The total number of such clauses is $3NK + N$. On the other hand, symmetries are broken among abstract actions with a weak ordering on them and by requiring $A^0 \leq A^1 \leq \cdots \leq A^{N-1}$. Abstract actions are ordered using the K bits for selected the affected features, for each $j \in \{1, \ldots, N-1\}$ and $k \in \{0, \ldots, K-1\}$:

$$\neg A^j_{1,0} \wedge \cdots \wedge \neg A^j_{1,k-1} \wedge A^j_{1,k} \rightarrow A^{j-1}_{1,0} \vee \cdots \vee A^{j-1}_{1,k}\,. \tag{6}$$

The number of clauses in (2)–(8) is $3NK + N + (N-1)K = N(1+4K) - K$.

The clauses needed for the F^k_t variables establish that no dummy or inexistent feature is selected, and enforce the strict ordering among features. The first group corresponds to the following clauses where d is an index of a dummy or inexistent feature:

$$\bigvee_{t:bit(t,d)} \neg F^k_t \;\vee \bigvee_{t:\neg bit(t,d)} F^k_t \tag{7}$$

where $bit(t,d)$ is a boolean constant for the t-th bit in the binary expansion of d. The ordering among selected featues is enforced with

$$df(k,0) \vee df(k,1) \vee \cdots \vee df(k,L) \tag{8}$$

for $k \in \{0, \ldots, K-2\}$ and $L = \log_2(F)$, and the boolean variables $df(k,t)$ satisfy

$$df(k,t) \rightarrow \neg F^k_t\,, \tag{9}$$

$$df(k,t) \rightarrow \neg F^{k+1}_t\,, \tag{10}$$

$$df(k,t) \wedge F^k_i \rightarrow F^{k+1}_i \qquad \text{for } i \in \{t+1, \ldots, L-1\}\,, \tag{11}$$

$$df(k,t) \wedge \neg F^k_i \rightarrow \neg F^{k+1}_i \qquad \text{for } i \in \{t+1, \ldots, L-1\}\,. \tag{12}$$

The total number of clauses in (7)–(12) is $K(D+U) + (K-1)(1 + 2L + L(L-1))$ where D and U are the number of dummy and inexistent features respectively. The total number of clauses in (2)–(12) is $Q(L,K,N) = N(1+4K) + K(D+U) + (K-1)(L^2+L) - 1$.

2.1 Value of selected features at states

We define the variable $\phi(k,i)$ that tells whether the selected feature F^k holds at state s_i in the sample; $\phi(k,i)$ can be defined in two equivalent ways:

$$\phi(k,i) \;\equiv\; \bigwedge_{\ell\in\mathcal{F}} [F^k = f_\ell] \rightarrow (m_{\ell i} > 0) \;\equiv\; \bigwedge_{\ell\in\mathcal{F}:m_{\ell i}=0} \neg[F^k = f_\ell] \;\equiv\; \bigwedge_{\ell\in\mathcal{F}:m_{\ell i}=0} \neg\Big(\bigwedge_t F_t^k \leftrightarrow bit(t,\ell)\Big) \tag{13}$$

$$\phi(k,i) \;\equiv\; \bigvee_{\ell\in\mathcal{F}} [F^k = f_\ell] \wedge (m_{\ell i} > 0) \;\equiv\; \bigvee_{\ell\in\mathcal{F}:m_{\ell i}>0} [F^k = f_\ell] \;\equiv\; \bigvee_{\ell\in\mathcal{F}:m_{\ell i}>0} \Big(\bigwedge_t F_t^k \leftrightarrow bit(t,\ell)\Big) \tag{14}$$

where $[F^k = f_\ell]$ denotes the formula that says the k-th selected feature is f_ℓ. Therefore, for each $k \in K$ and $i \in \mathcal{M}$, the variable $\phi(k,i)$ is defined with the following implications:

$$\phi(k,i) \;\rightarrow\; \Big(\bigvee_{t:bit(t,\ell)} \neg F_t^k\Big) \vee \Big(\bigvee_{t:\neg bit(t,\ell)} F_t^k\Big), \quad \text{for each } \ell \text{ such that } m_{\ell i} = 0, \tag{15}$$

$$\phi(k,i) \;\leftarrow\; \Big(\bigwedge_{t:bit(t,\ell)} F_t^k\Big) \wedge \Big(\bigwedge_{t:\neg bit(t,\ell)} \neg F_t^k\Big), \quad \text{for each } \ell \text{ such that } m_{\ell i} > 0. \tag{16}$$

These account for exactly SKF clauses where S is the number of states in $\mathcal{M}$ and F is the number of features in $\mathcal{F}$. Likewise, we define the variable $\phi^*(k,i)$ that tells whether the selected feature F^j *does not* hold at s_i:

$$\phi^*(k,i) \;\equiv\; \bigwedge_{\ell\in\mathcal{F}} [F^k = f_\ell] \rightarrow (m_{\ell i} = 0) \;\equiv\; \bigwedge_{\ell\in\mathcal{F}:m_{\ell i}>0} \neg[F^k = f_\ell] \;\equiv\; \bigwedge_{\ell\in\mathcal{F}:m_{\ell i}>0} \neg\Big(\bigwedge_t F_t^k \leftrightarrow bit(t,\ell)\Big) \tag{17}$$

$$\phi^*(k,i) \;\equiv\; \bigvee_{\ell\in\mathcal{F}} [F^k = f_\ell] \wedge (m_{\ell i} = 0) \;\equiv\; \bigvee_{\ell\in\mathcal{F}:m_{\ell i}=0} [F^k = f_\ell] \;\equiv\; \bigvee_{\ell\in\mathcal{F}:m_{\ell i}=0} \Big(\bigwedge_t F_t^k \leftrightarrow bit(t,\ell)\Big) \tag{18}$$

with the following SKF clauses:

$$\phi^*(k,i) \;\rightarrow\; \Big(\bigvee_{t:bit(t,\ell)} \neg F_t^k\Big) \vee \Big(\bigvee_{t:\neg bit(t,\ell)} F_t^k\Big), \quad \text{for each } \ell \text{ such that } m_{\ell i} > 0, \tag{19}$$

$$\phi^*(k,i) \;\leftarrow\; \Big(\bigwedge_{t:bit(t,\ell)} F_t^k\Big) \wedge \Big(\bigwedge_{t:\neg bit(t,\ell)} \neg F_t^k\Big), \quad \text{for each } \ell \text{ such that } m_{\ell i} = 0. \tag{20}$$

2.2 Soundness

We express soundness with the following formulas, one for each state s in the sample $\mathcal{M}$, and for each $j \in \{0,\ldots,N-1\}$:

$$app(A^j,[s]) \rightarrow \bigvee_{t=\langle s,a,s'\rangle\in\mathcal{M}} t \sim \langle [s], A^j, res(A^j,[s])\rangle \tag{21}$$

where $app(A^j,[s])$ means that the abstract action denoted by A^j is applicable at the abstract state $\tilde{s}$ such that $s \sim \tilde{s}$, $res(A^j,[s])$ denotes the abstract state that results of applying A^j in $\tilde{s}$, and $\tilde{t} = \langle [s], A^j, res(A^j,[s])\rangle$ is the abstract transition from $\tilde{s}$ to $\tilde{s}' = res(A^j,[s])$ induced by A^j.

For $s_i = s$, the formula $app(A^j,[s])$ is captured with:

$$app(A^j,[s]) \;\equiv\; \bigwedge_k \Big(A^j_{4,k} \rightarrow \phi(k,i)\Big) \;\wedge\; \bigwedge_k \Big(A^j_{3,k} \wedge \neg A^j_{4,k} \rightarrow \phi^*(k,i)\Big) \;\equiv\; \bigwedge_k B_1(j,i,k) \wedge B_2(j,i,k) \tag{22}$$

where the auxiliary variables $B_1(j,i,k)$ and $B_2(j,i,k)$ are given by:

$$B_1(j,i,k) \wedge A^j_{4,k} \rightarrow \phi(k,i)\,, \tag{23}$$

$$\neg A^j_{4,k} \rightarrow B_1(j,i,k)\,, \tag{24}$$
$$\phi(k,i) \rightarrow B_1(j,i,k)\,, \tag{25}$$

$$B_2(j,i,k) \wedge A^j_{3,k} \wedge \neg A^j_{4,k} \rightarrow \phi^*(k,i)\,, \tag{26}$$
$$\neg A^j_{3,k} \rightarrow B_2(j,i,k)\,, \tag{27}$$
$$A^j_{4,k} \rightarrow B_2(j,i,k)\,, \tag{28}$$
$$\phi^*(k,i) \rightarrow B_2(j,i,k)\,. \tag{29}$$

In total, we need $7NSK$ clauses to define the auxiliary B variables.

Let us focus on the consequent of the implication that defines soundness. We need an expression for $t \sim \langle [s], A^j, res(A^j,[s])\rangle$ where $t = \langle s,a,s'\rangle$ is a transition in $\mathcal{M}$, A^j is an abstract action, and $res(A^j,[s])$ denotes the abstract state that results of applying A^j on $[s]$. By definition, $\langle s,a,s'\rangle \sim \langle \tilde{s},\tilde{a},\tilde{s}'\rangle$ iff 1) $s \sim \tilde{s}$, 2) $s' \sim \tilde{s}'$, and 3) the numeric features change in both transitions in a compatible way. In our case, $t = \langle s_i, a, s_\ell\rangle$ and $\tilde{t} = \langle [s_i], A^j, res(A^j,[s_i])\rangle$ and (1) is always satisfied. For (2), we use the auxiliary variable $RES(i,\ell,j)$ to encode $s_\ell \sim res(A^j,[s_i])$:

$$RES(i,\ell,j) \;\equiv\; \bigwedge_k \Big(A^j_{2,k} \rightarrow \phi(k,\ell) \Big) \;\wedge\; \bigwedge_k \Big(A^j_{1,k} \wedge \neg A^j_{2,k} \wedge \neg[F^k \text{ is numeric}] \rightarrow \phi^*(k,\ell) \Big) \;\wedge \tag{30}$$
$$\bigwedge_k \Big(\neg A^j_{1,k} \wedge \phi(k,i) \rightarrow \phi(k,\ell) \Big) \;\wedge\; \bigwedge_k \Big(\neg A^j_{1,k} \wedge \phi^*(k,i) \rightarrow \phi^*(k,\ell) \Big)\,. \tag{31}$$

Since $RES(i,\ell,j)$ only appears in the consequent of the implications defining soundness, it is enough to enforce the following implications:

$$RES(i,\ell,j) \wedge A^j_{2,k} \rightarrow \phi(k,\ell)\,, \tag{32}$$
$$RES(i,\ell,j) \wedge A^j_{1,k} \wedge \neg A^j_{2,k} \wedge \neg[F^k \text{ is numeric}] \rightarrow \phi^*(k,\ell)\,, \tag{33}$$
$$RES(i,\ell,j) \wedge \neg A^j_{1,k} \wedge \phi(k,i) \rightarrow \phi(k,\ell)\,, \tag{34}$$
$$RES(i,\ell,j) \wedge \neg A^j_{1,k} \wedge \phi^*(k,i) \rightarrow \phi^*(k,\ell)\,. \tag{35}$$

There are 4 implications for each transition $t = \langle s_i, a, s_\ell\rangle$ in $\mathcal{M}$, each $j \in \{0,\ldots,N-1\}$, and each $k \in \{0,\ldots,K-1\}$.

The final requirement to enforce $t \sim \tilde{t}$ is that the numerical features change in both transitions in a compatible manner. For each numerical feature f, if f is increased (resp. decreased) by A^j, then f increases (resp. decreases) in the transition from s to s'. We use auxiliary variables $INC(k,i,\ell)$ (resp. $DEC(k,i,\ell)$) to denote that feature k increases (resp. decreases) in the transition $\langle s_i, a, s_\ell\rangle$ in $\mathcal{M}$.[2] The $INC(k,i,\ell)$ variable can be expressed in two different ways:

$$INC(k,i,\ell) \;\equiv\; \bigwedge_{j\in\mathcal{F}} [F^k = f_j] \rightarrow (m_{ji} < m_{j\ell}) \;\equiv\; \bigwedge_{j\in\mathcal{F}: m_{ji} \geq m_{j\ell}} \neg[F^k = f_j] \tag{36}$$
$$\equiv \bigwedge_{j\in\mathcal{F}: m_{ji} \geq m_{j\ell}} \neg\Big(\bigwedge_t F^k_t \leftrightarrow bit(t,j) \Big)\,, \tag{37}$$
$$INC(k,i,\ell) \;\equiv\; \bigvee_{j\in\mathcal{F}} [F^k = f_j] \wedge (m_{ji} < m_{j\ell}) \;\equiv\; \bigvee_{j\in\mathcal{F}: m_{ji} < m_{j\ell}} [F^k = f_j] \tag{38}$$
$$\equiv \bigvee_{j\in\mathcal{F}: m_{ji} < m_{j\ell}} \Big(\bigwedge_t F^k_t \leftrightarrow bit(t,j) \Big)\,. \tag{39}$$

[2] **** CHECK: Can we have two transitions with equal source and destination but with different actions? Does it matter? It seems that it doesn't matter since whether f increases or decreases in a transition only depent of the values of f at the states. ****

We therefore use the following implications to define $INC(k,i,\ell)$:

$$INC(k,i,\ell) \;\rightarrow\; \left(\bigvee_{t:bit(t,j)} \neg F_t^k\right) \vee \left(\bigvee_{t:\neg bit(t,j)} F_t^k\right), \quad \text{for each } j \text{ such that } m_{ji} \geq m_{j\ell}, \tag{40}$$

$$INC(k,i,\ell) \;\leftarrow\; \left(\bigwedge_{t:bit(t,j)} F_t^k\right) \wedge \left(\bigwedge_{t:\neg bit(t,j)} \neg F_t^k\right), \quad \text{for each } j \text{ such that } m_{ji} < m_{j\ell}. \tag{41}$$

The $DEC(k,i,\ell)$ variables are defined analogously with the implications:

$$DEC(k,i,\ell) \;\rightarrow\; \left(\bigvee_{t:bit(t,j)} \neg F_t^k\right) \vee \left(\bigvee_{t:\neg bit(t,j)} F_t^k\right), \quad \text{for each } j \text{ such that } m_{ji} \leq m_{j\ell}, \tag{42}$$

$$DEC(k,i,\ell) \;\leftarrow\; \left(\bigwedge_{t:bit(t,j)} F_t^k\right) \wedge \left(\bigwedge_{t:\neg bit(t,j)} \neg F_t^k\right), \quad \text{for each } j \text{ such that } m_{ji} > m_{j\ell}. \tag{43}$$

Therefore, there are $2TK$ such auxiliaries variables that are defined with $2TFK$ implications. The following auxiliary variables capture $\langle s_i, a, s_\ell\rangle \sim \langle [s_i], A^j, res(A^j,[s_i])\rangle$:

$$MATCH(i,\ell,j) \;\equiv\; RES(i,\ell,j) \wedge \tag{44}$$

$$\bigwedge_k \Big([F^k \text{ is numeric}] \rightarrow (A^j_{2,k} \leftrightarrow INC(k,i,\ell)) \Big) \wedge \tag{45}$$

$$\bigwedge_k \Big(A^j_{1,k} \wedge [F^k \text{ is numeric}] \rightarrow (\neg A^2_{2,k} \leftrightarrow DEC(k,i,\ell)) \Big). \tag{46}$$

As these variables only appear in the consequent of implications, it is sufficient to only enforce:

$$MATCH(i,\ell,j) \rightarrow RES(i,\ell,j)\,, \tag{47}$$

$$MATCH(i,\ell,j) \wedge [F^k \text{ is numeric}] \wedge A^j_{2,k} \rightarrow INC(k,i,\ell)\,, \tag{48}$$

$$MATCH(i,\ell,j) \wedge [F^k \text{ is numeric}] \wedge INC(k,i,\ell) \rightarrow A^j_{2,k}\,, \tag{49}$$

$$MATCH(i,\ell,j) \wedge [F^k \text{ is numeric}] \wedge A^j_{1,k} \wedge \neg A^j_{2,k} \rightarrow DEC(k,i,\ell)\,, \tag{50}$$

$$MATCH(i,\ell,j) \wedge [F^k \text{ is numeric}] \wedge DEC(k,i,\ell) \rightarrow A^j_{1,k}\,, \tag{51}$$

$$MATCH(i,\ell,j) \wedge [F^k \text{ is numeric}] \wedge DEC(k,i,\ell) \rightarrow \neg A^j_{2,k}\,. \tag{52}$$

There are $TN + 5TNK$ such implications.

Finally, soundness is enforced with SN implications, one for each state s_i and $j \in \{0,\ldots,N-1\}$:

$$\left(\bigwedge_k B_1(j,i,k) \wedge B_2(j,i,k)\right) \;\rightarrow \bigvee_{t=\langle s_i,a,s_\ell\rangle} MATCH(i,\ell,j)\,. \tag{53}$$

2.3 Completeness

For each transition $t = \langle s, a, s'\rangle$ in $\mathcal{M}$, there is an abstract transition $\tilde{t}$ such that $t \sim \tilde{t}$; that is, for each transition $t = \langle s_i, a, s_\ell\rangle$ in $\mathcal{M}$:

$$\textstyle\bigvee_j app(A^j,[s_i]) \wedge t \sim \langle [s_i], A^j, res(A^j,[s_i])\rangle\,. \tag{54}$$

Since $MATCH(i,\ell,j)$ implies $\langle s_i, a, s_\ell\rangle \sim \langle [s_i], A^j, res(A^j,[s_i])\rangle$, completeness can be capture with

$$\textstyle\bigvee_j APPMATCH(i,\ell,j)\,, \tag{55}$$

one for each transition $\langle s_i, a, s_\ell\rangle$ in $\mathcal{M}$, where the auxiliary variables $APPMATCH(i,\ell,j)$ satisfy

$$APPMATCH(i,\ell,j) \rightarrow MATCH(i,\ell,j)\,, \tag{56}$$

$$APPMATCH(i,\ell,j) \rightarrow B_1(j,i,k)\,, \tag{57}$$

$$APPMATCH(i,\ell,j) \rightarrow B_2(j,i,k) \tag{58}$$

for $k = 0,\ldots,K-1$.