Formalized Theorems from the Paper "A Coalgebraic Decision Procedure for WS1S"

Dmitriy Traytel

January 19, 2015

lemma
fixes I :: *interp*
and x y X :: *nat*
and φ ψ :: *formula*
shows
$I \models T \longleftrightarrow True$
$I \models F \longleftrightarrow False$
$I \models (FO\ x) \longleftrightarrow I[x]_1 \neq \{\}$
$I \models (x < y) \longleftrightarrow Min\ (I[x]_1) < Min\ (I[y]_1) \wedge I[x]_1 \neq \{\} \wedge I[y]_1 \neq \{\}$
$I \models (x \in X) \longleftrightarrow Min\ (I[x]_1) \in I[X]_2 \wedge I[x]_1 \neq \{\} \wedge finite\ (I[X]_2)$
$I \models (\neg\ \varphi) \longleftrightarrow \neg\ (I \models \varphi)$
$I \models (\varphi \vee \psi) \longleftrightarrow (I \models \varphi \vee I \models \psi)$
$I \models (FAnd\ \varphi\ \psi) \longleftrightarrow (I \models \varphi \wedge I \models \psi)$
$I \models (\exists_1\ \varphi) \longleftrightarrow (\exists P.\ finite\ P \wedge P::_{1as2} I \models \varphi)$
$I \models (\exists_2\ \varphi) \longleftrightarrow (\exists P.\ finite\ P \wedge P::_2 I \models \varphi)$
by (*auto 0 2 simp*: *Let_def fMin.rep_eq fmember.rep_eq*
fset_inverse intro: *exI*[*of* _ *fset P* **for** *P*])

lemma
fixes I :: *interp*
and x y X :: *nat*
and φ ψ :: *formula*
shows
$I \models_< T \longleftrightarrow True$
$I \models_< F \longleftrightarrow False$
$I \models_< (FO\ x) \longleftrightarrow I[x]_1 \neq \{\}$
$I \models_< (x < y) \longleftrightarrow Min\ (I[x]_1) < Min\ (I[y]_1) \wedge I[x]_1 \neq \{\} \wedge I[y]_1 \neq \{\}$
$I \models_< (x \in X) \longleftrightarrow Min\ (I[x]_1) \in I[X]_2 \wedge I[x]_1 \neq \{\} \wedge finite\ (I[X]_2)$
$I \models_< (\neg\ \varphi) \longleftrightarrow \neg\ (I \models_< \varphi)$
$I \models_< (\varphi \vee \psi) \longleftrightarrow (I \models_< \varphi \vee I \models_< \psi)$
$I \models_< (FAnd\ \varphi\ \psi) \longleftrightarrow (I \models_< \varphi \wedge I \models_< \psi)$
$I \models_< (\exists_1\ \varphi) \longleftrightarrow (\exists P.\ (\forall p \in P.\ p <\# I) \wedge P::_{1as2} I \models_< \varphi)$
$I \models_< (\exists_2\ \varphi) \longleftrightarrow (\exists P.\ (\forall p \in P.\ p <\# I) \wedge P::_2 I \models_< \varphi)$
by (*auto 0 2 simp*: *Let_def fMin.rep_eq fmember.rep_eq*
len_leq_iff Abs_fset_inverse bounded_nat_set_is_finite fset_inverse
elim: *exI*[*of* _ *Abs_fset P* **for** *P*, *OF conjI*, *rotated*])

lemma
fixes I :: *interp*
and x y X :: *nat*
and φ ψ :: *formula*
shows
$I \Models T \longleftrightarrow True$
$I \Models F \longleftrightarrow False$
$I \Models (FO\ x) \longleftrightarrow I[x]_1 \neq \{\}$

$I \models (x < y) \longleftrightarrow Min\ (I[x]_1) < Min\ (I[y]_1) \wedge I[x]_1 \neq \{\} \wedge I[y]_1 \neq \{\}$
$I \models (x \in X) \longleftrightarrow Min\ (I[x]_1) \in I[X]_2 \wedge I[x]_1 \neq \{\} \wedge finite\ (I[X]_2)$
$I \models (\neg\ \varphi) \longleftrightarrow \neg\ (I \models \varphi)$
$I \models (\varphi \vee \psi) \longleftrightarrow (I \models \varphi \vee I \models \psi)$
$I \models (FAnd\ \varphi\ \psi) \longleftrightarrow (I \models \varphi \wedge I \models \psi)$
$I \models (\exists_1\ \varphi) \longleftrightarrow (\exists\, p.\ p ::_1 I \models \varphi)$
$I \models (\exists_2\ \varphi) \longleftrightarrow (\exists\, P.\ finite\ P \wedge P ::_2 I \models \varphi)$
by (*auto simp add*: *Let_def fMin.rep_eq fmember.rep_eq*)

lemma
fixes I :: *interp*
and $x\ y\ X$ :: *nat*
and $\varphi\ \psi$ :: *formula*
shows
$I \models_< T \longleftrightarrow True$
$I \models_< F \longleftrightarrow False$
$I \models_< (FO\ x) \longleftrightarrow I[x]_1 \neq \{\}$
$I \models_< (x < y) \longleftrightarrow Min\ (I[x]_1) < Min\ (I[y]_1) \wedge I[x]_1 \neq \{\} \wedge I[y]_1 \neq \{\}$
$I \models_< (x \in X) \longleftrightarrow Min\ (I[x]_1) \in I[X]_2 \wedge I[x]_1 \neq \{\} \wedge finite\ (I[X]_2)$
$I \models_< (\neg\ \varphi) \longleftrightarrow \neg\ (I \models_< \varphi)$
$I \models_< (\varphi \vee \psi) \longleftrightarrow (I \models_< \varphi \vee I \models_< \psi)$
$I \models_< (\exists_1\ \varphi) \longleftrightarrow (\exists\, p <\# I.\ p ::_1 I \models_< \varphi)$
$I \models_< (\exists_2\ \varphi) \longleftrightarrow (\exists\, P.\ (\forall\, p \in P.\ p <\# I) \wedge P ::_2 I \models_< \varphi)$
by (*auto simp add*: *Let_def fMin.rep_eq fmember.rep_eq*)

abbreviation *bisimilar* (**infix** $\sim$ *65*) **where**
$L \sim K \equiv (\exists\, R.\ R\ L\ K \wedge (\forall\, L'\ K'.\ R\ L'\ K' \longrightarrow$
$((] \in L' \longleftrightarrow [] \in K') \wedge (\forall\, a.\ R\ (L')_a\ (K')_a))))$

theorem *Theorem1*:
fixes $L\ K$ :: *'a language*
shows $L \sim K \Longrightarrow L = K$
by (*coinduction arbitrary*: $K\ L$) *auto*

lemma *Lemma2*:
fixes Σ :: *'a list*
and L :: *'t* $\Rightarrow$ *'a language*
and L' :: *'s* $\Rightarrow$ *'a language*
and ι :: *'s* $\Rightarrow$ *'t*
and δ :: *'a* $\Rightarrow$ *'t* $\Rightarrow$ *'t*
and o :: *'t* $\Rightarrow$ *bool*
and *wf* :: *'t* $\Rightarrow$ *bool*
assumes $\bigwedge s\ w.\ wf\ s \Longrightarrow w \in L\ s \Longrightarrow w \in \Sigma^*$
and $\bigwedge t.\ L\ (\iota\ t) = L'\ t$
and $\bigwedge s\ a.\ wf\ s \Longrightarrow a \in set\ \Sigma \Longrightarrow wf\ (\delta\ a\ s)$
and $\bigwedge s\ a.\ wf\ s \Longrightarrow a \in set\ \Sigma \Longrightarrow L\ (\delta\ a\ s) = (L\ s)_a$
and $\bigwedge s.\ wf\ s \Longrightarrow$ o $s \longleftrightarrow [] \in L\ s$
and $\bigwedge s.\ wf\ s \Longrightarrow finite\ \{fold\ \delta\ w\ s\ |w.\ w \in \Sigma^*\}$
and $wf\ (\iota\ s)\ wf\ (\iota\ s')$
shows $bisim\ wf\ \Sigma\ \iota\ \delta$ o $s\ s' \longleftrightarrow L'\ s = L'\ s'$
proof −
interpret *D*: *DFA* $\Sigma\ \iota\ \delta$ o $wf\ \lambda s.\ wf\ (\iota\ s)\ L\ L'$
using *assms* **by** *unfold_locales auto*
show $bisim\ wf\ \Sigma\ \iota\ \delta$ o $s\ s' \longleftrightarrow L'\ s = L'\ s'$ **by** (*auto intro*: *D.soundness D.completeness assms*)
qed

lemma *Theorem3*:
fixes φ :: *formula*
and I :: *interp*
and a :: *bool list* $\times$ *bool list*
assumes *wf* $(\#_V\ I)\ \varphi$
and $\#_V\ I = |a|$
shows $I \models \delta\ a\ \varphi \longleftrightarrow CONS\ a\ I \models \varphi$
and $I \models_< \delta\ a\ \varphi \longleftrightarrow CONS\ a\ I \models_< \varphi$
by (*rule WS1S.satisfies_lderiv*[*OF assms*], *rule WS1S.satisfies_bounded_lderiv*[*OF assms*])

lemma *Theorem4*:
fixes φ :: *formula*
shows *finite* $\{\ |fold\ \delta\ xs\ \varphi|_{ACI}\ |\ xs.\ True\}$
by (*blast intro*: *WS1S.finite_fold_deriv*)

lemma *Example1*:
shows $|\delta\ ([False], [])\ (Ex_2\ (0 \in 0))|_{ACI} = Ex_2\ (0 \in 0)$
and $|\delta\ ([True], [])\ (Ex_2\ (0 \in 0))|_{ACI} = Ex_2\ (F \vee T)$
and $|\delta\ ([False], [])\ (Ex_2\ (F \vee T))|_{ACI} = Ex_2\ (F \vee T)$
and $|\delta\ ([True], [])\ (Ex_2\ (F \vee T))|_{ACI} = Ex_2\ (F \vee T)$
by *eval*+

lemma *Theorem5*:
fixes φ :: *formula*
shows *finite* $\{\ |fold\ \varrho\ xs\ \varphi|_{ACI}\ |\ xs.\ True\}$
by (*blast intro*: *WS1S.finite_fold_deriv*)

lemma *Theorem6*:
fixes φ :: *formula*
and I :: *interp*
and a :: *bool list* $\times$ *bool list*
assumes *wf* $(\#_V\ I)\ \varphi$
and $\#_V\ I = |a|$
shows $I \models_< \varrho\ a\ \varphi \longleftrightarrow SNOC\ a\ I \models_< \varphi$
by (*rule WS1S.satisfies_bounded_rderiv*[*OF assms*])

lemma *Theorem71*:
fixes φ :: *formula*
and I :: *interp*
assumes *wf* $(\#_V\ I)\ \varphi$
and $\#I = 0$
shows $\mathrm{o}_<\ \varphi \longleftrightarrow I \models_< \varphi$
using *assms* **by** (*auto simp*: *WS1S.nullable_satisfies_bounded*)

lemma *Theorem72*:
fixes φ :: *formula*
and I :: *interp*
assumes *wf* $(\#_V\ I)\ \varphi$
shows $I \models_< futurize\ (\#_V\ I)\ \varphi \longleftrightarrow$
$(\exists k.\ (SNOC\ (zero\ (\#_V\ I))\ \hat{}\hat{}\ k)\ I \models_< \varphi)$
using *assms* **by** (*auto simp*: *WS1S.satisfies_bounded_fut*)

lemma *Theorem73*:
fixes φ :: *formula*
and I :: *interp*
assumes *wf* $(\#_V\ I)\ \varphi$
shows $I \models_< \lfloor\varphi\rfloor_{(\#_V\ I)} \longleftrightarrow I \models \varphi$
using *assms* **by** (*auto simp*: *WS1S.finalize_satisfies*)

lemma *Theorem74*:
fixes φ :: *formula*
and I :: *interp*
assumes *wf* $(\#_V\ I)\ \varphi$
and $\#I = 0$
shows o $(\#_V\ I)\ \varphi \longleftrightarrow I \models \varphi$
using *assms* **by** (*auto simp*: *WS1S.final_satisfies*)

lemma *language_def*:
$L\ n\ \varphi = \{enc\ I \mid I.\ I \Turnstile \varphi \wedge (\forall x{\in}FOV\ \varphi.\ I[x]_1 \neq \{\}) \wedge \#_V\ I = n\}$
$L_<\ n\ \varphi = \{enc\ I \mid I.\ I \Turnstile_< \varphi \wedge (\forall x{\in}FOV\ \varphi.\ I[x]_1 \neq \{\}) \wedge \#_V\ I = n\}$
unfolding *WS1S.language_def WS1S.language$_b$_def sat_alt sat$_b$_alt* **by** *simp_all*

lemma *Theorem8*:
fixes $\varphi\ \psi$ :: *formula*
and n :: *interp_size*
shows *eqv* $n\ \varphi\ \psi \Longrightarrow L\ n\ \varphi = L\ n\ \psi$
and *eqv*$_<$ $n\ \varphi\ \psi \Longrightarrow L_<\ n\ \varphi = L_<\ n\ \psi$
unfolding *check_eqv_def bounded_check_eqv_def*
by (*drule WS1S.soundness*, *erule injD*[*OF bij_is_inj*[*OF to_language_bij*]])
(*drule WS1S.bounded.soundness*, *erule injD*[*OF bij_is_inj*[*OF to_language_bij*]])

lemma *Example2*:
shows *eqv* $\langle 1,\ 0\rangle$ $(Ex_2\ (0 \in 0))$ $(FO\ 0)$
by *eval*